



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

## EXTENSIÓN DE LA PLATAFORMA U-CALENDAR

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN  
COMPUTACIÓN

CRISTIAN ARIEL SÁNCHEZ ARÁNGUIZ

PROFESOR GUÍA:  
SERGIO OCHOA DE LORENZI

MIEMBROS DE LA COMISIÓN:  
ALEXANDRE HENRI BERGEL  
JUAN ÁLVAREZ RUBIO

SANTIAGO DE CHILE

2022

# Resumen

Uno de los grandes desafíos a enfrentar en la Universidad de Chile, incluyendo al Departamento de Ciencias de la Computación (DCC), es la gestión de la carga académica estudiantil. La mala gestión de este aspecto, desencadenó diversas manifestaciones estudiantiles durante el año 2019, lo que llevó a repensar la forma en la que se aborda este tema. A nivel del DCC se tomaron varias acciones, que trajeron aparejado una gran carga de trabajo para la secretaría docente, la cual se encarga de velar por una distribución razonable de la carga académica estudiantil.

Con el fin de simplificar la labor de balancear la carga académica asociada a las evaluaciones de cursos de la ingeniería civil en computación, en el 2020 nace UCalendar. Esta es una aplicación que permitía manejar usuarios de diverso tipo, definirles un cierto rol en la aplicación, crear ramos, cursos y semestres, todo esto de forma manual. Además, permitía visualizar en un mapa de calor la carga académica asociada a las evaluaciones del semestre. A pesar de los avances que permitía esta aplicación para la gestión de evaluaciones, su uso aún requería un alto grado de trabajo manual y repetitivo. Además, había varias inconsistencias en el sistema que hacían necesario realizar varias modificaciones y extensiones para su puesta en producción. De esta manera, el objetivo final de este trabajo de título es dotar a UCalendar de la robustez y funcionalidades mínimas para que el sistema sea utilizable de forma fácil y agradable para la secretaría docente, los profesores y auxiliares, y también para los alumnos.

Para ello se priorizaron las funcionalidades esenciales para el funcionamiento del sistema, en particular aquellas que permiten cumplir el objetivo antes mencionado. Entre las funcionalidades esenciales está la alimentación automática de cursos, profesores y alumnos desde la base de datos de Docencia del DCC. Este proceso permite disminuir en gran medida el trabajo realizado, y facilita la gestión de las evaluaciones, para así mejorar su distribución en el semestre. Por su parte, los estudiantes acceden a información personalizada respecto a las evaluaciones de sus cursos, y de esa forma pueden mejorar su organización.

Para evaluar la solución implementada fue necesario hacer un *recorrido cognitivo* constante durante el proceso de implementación, con el fin de obtener feedback inmediato sobre las funcionalidades entregadas por el sistema. Además, se realizó un recorrido cognitivo en la etapa final de la aplicación para detectar aquellos puntos donde se necesita mejorar, y detectar las dificultades remanentes para los usuarios. Seguido de esto, se aplicó un cuestionario SUS para comprender, desde la perspectiva de estos usuarios, el nivel de usabilidad del sistema y las oportunidades de mejora.

Tal como se planteó en un inicio, este trabajo de título representó un cambio importante en la usabilidad que poseía el sistema UCalendar, disminuyendo el trabajo repetitivo y simplificando las tareas de los usuarios. Además, se consideró la opinión de los estudiantes sobre su carga académica. Como trabajo a futuro, se plantea conocer el efecto que el uso de este sistema tiene en los estudiantes, y analizar posibles mejoras al mismo.

# Tabla de contenido

<b>Capítulo 1</b>	<b>1</b>
<b>Introducción</b>	<b>1</b>
1.1 Aspectos generales	1
1.2 Justificación del trabajo	2
1.3 Objetivos de la memoria	3
1.4 Descripción general de la solución	4
1.5 Estructura del documento	4
<b>Capítulo 2</b>	<b>6</b>
<b>Análisis de UCalendar</b>	<b>6</b>
2.1 Contextualización del proyecto	6
2.2 Estructura del sistema legado	7
2.3 Funcionalidades del sistema legado	8
2.4 Mapa de navegación del sistema	9
2.5 Interfaces de usuario del sistema legado	11
2.6 Público objetivo y roles de usuario soportados	16
<b>Capítulo 3</b>	<b>17</b>
<b>Reingeniería del sistema</b>	<b>17</b>
3.1 Definición de los requisitos del nuevo sistema	17
3.1.1 Alimentación desde la base de datos de docencia	19
3.1.2 Monitoreo y sugerencia de evaluaciones	20
3.1.3 FeedBack a estudiantes: gestión de notificaciones	21
3.2 Modificaciones a la estructura del sistema	22
3.3 Modificaciones a la navegación e interfaces del sistema	23
3.3.1 Navegación del sistema	23
3.3.2 Diseño de las interfaces de usuario	26
3.4 Refactoring y debugging del sistema legado	29
3.5 Herramientas utilizadas	30
<b>Capítulo 4</b>	<b>31</b>
<b>Solución implementada</b>	<b>31</b>
4.1 Modificaciones de interfaces de usuario	31
4.1.1 Vista de calendario	31
4.1.2 Vista de semestres	32
4.1.3 Vista de evaluaciones	33
4.1.4 Vista de evaluaciones 2	34
4.1.5 Vista de ramos	35

4.2 Servicio de alimentación de datos	36
4.3 Servicio de envío de notificaciones a estudiantes	37
4.4 Mejora de la gestión de evaluaciones	38
4.4.1 Sistema de notificaciones de alerta	38
4.4.2 Sugerencia de evaluaciones	38
4.4.3 Posponer el calendario académico	38
4.4.4 Modificación del mapa de calor	39
4.4.4.1 Percepción de cantidad de evaluaciones para un día.	39
4.4.4.2 Percepción de carga aceptable para un día	40
4.4.4.3 Percepción de carga aceptable para una semana	41
4.4.4.4 Analisis de resultados de la encuesta	43
4.5 Servicio de autenticación de usuarios y asignación de roles	43
<b>Capítulo 5</b>	<b>47</b>
<b>Evaluación de la solución</b>	<b>47</b>
5.1 Descripción del proceso de evaluación	47
5.1.1 Recorrido Cognitivo	47
5.1.2 Resultados del Recorrido Cognitivo	49
5.2 Evaluación de la usabilidad de la aplicación	49
5.2.1 Descripción de la System Usability Scale	49
5.2.2 Resultados de la evaluación de usabilidad	50
<b>Capítulo 6</b>	<b>52</b>
<b>Conclusiones y trabajo a futuro</b>	<b>52</b>
<b>Bibliografía</b>	<b>53</b>
<b>Glosario</b>	<b>54</b>

# Capítulo 1

## Introducción

### 1.1 Aspectos generales

La carga académica ha sido uno de los grandes desafíos a enfrentar para mejorar la calidad de vida estudiantil. Es por esto que en el Departamento de Ciencias de la Computación de la Universidad de Chile (DCC) nace el proyecto UCalendar, como una posible solución a este problema.

UCalendar es un software que inicia su desarrollo en el curso de “CC5401: Ingeniería de Software II”, el cual busca planificar y gestionar la carga de evaluaciones del estudiantado, en base a bloques semestrales. Esta información se representa mediante un mapa de calor, que permite a los docentes reconocer las fechas de mayor sobrecarga de evaluaciones para su alumnado (principalmente controles y entregas de tareas). De esa manera, los docentes pueden reagendar la fecha de sus evaluaciones en caso de sobrecarga, mejorando la distribución de éstas durante el semestre. Además de lo anterior, este software ayuda a la Secretaría Docente a controlar situaciones académicas perjudiciales para los alumnos; por ejemplo, tener tres controles de cursos obligatorios el mismo día.

La herramienta ha sido desarrollada durante los semestres de Otoño y Primavera del año 2020. A pesar de que el sistema cuenta con funciones suficientes para ser utilizable, aún falta agregar servicios, realizar ajustes y validaciones, que le permitan ser considerado un producto mínimo viable, y por lo tanto, entrar en fase de producción. Particularmente, existían cuatro aspectos a mejorar, los cuales se describen a continuación:

1. *Alimentación de datos del sistema:* El primer aspecto a mejorar es la carga de información acerca de los cursos que se dictan en un cierto semestre, y sus respectivas fechas de controles. Hoy en día esta información es generada manualmente por el área de docencia. Debido a esto, resulta relevante la carga de trabajo que significa añadir nueva información al sistema y el tiempo requerido para hacerlo; además de los potenciales errores de digitación debido a la gran carga manual, y del esfuerzo de coordinación con los profesores de los diferentes cursos que esto conlleva.
2. *Actualización de Login:* En la iteración anterior del desarrollo de este sistema se realizó una integración de UCalendar con el sistema de Pasaporte UChile. Sin embargo, en el corto plazo se dispuso de un “single sign on” para las aplicaciones del DCC (Portal DCC), el cual realiza autenticación contra Pasaporte Uchile, por lo que se requirió que el login de la aplicación se realizaré a través de este nuevo sistema.

3. *Entrega de información a los estudiantes:* Anteriormente los estudiantes contaban con una interfaz pública de consulta de controles, donde ellos indican los cursos que están cursando, y el sistema les genera un cronograma (con información representada tipo mapa de calor) con las evaluaciones de dichos cursos. Aunque este servicio está funcional, el alumno debe consultar sistemas externos para generar un calendario asociado a sus cursos actuales, además de regenerar este calendario en caso de cambios en la fecha de evaluación. En relación con este aspecto, se requirió que el sistema fuese proactivo, y amigable con este tipo de usuarios..
4. *Mejora de la gestión de fechas de evaluaciones:* Para administrar la carga académica, la Secretaría Docente del Departamento debe verificar de manera manual los mapas de calor de cada uno de los semestres actuales. Como mejora a esta forma de operar se requirió la implementación de un sistema de notificaciones que permitiese alertar a esta unidad sobre aquellas fechas y evaluaciones relevantes a monitorear.

## 1.2 Justificación del trabajo

Para dar respuesta a las limitaciones que existen en el sistema UCalendar legado, en esta memoria se analizan e implementan una serie de servicios que se describen a continuación.

1. *Inyector de datos al sistema:* La carga de usuarios (particularmente, los estudiantes, ayudantes, auxiliares y profesores de los cursos), en conjunto con los ramos, cursos y semestres, se realiza ahora de forma automática desde la Base de Datos de Docencia, evitando el trabajo manual a través de formularios. Este trabajo representa un desafío, ya que se investiga y prueba si dicho sistema cuenta con end-points destinados para obtener la información requerida, se solicita la incorporación de nueva información y finalmente se procede a identificar aquella información que debiese ser obtenida manualmente.
2. *Envío de notificaciones a estudiantes:* En instancias previas los estudiantes no tienen mayores interacciones con el sistema, más allá de la visualización de diversas materias en un mapa de calor. Para mejorar la concientización de los estudiantes respecto a las actividades evaluativas que tienen por delante, así como cambios en las fechas estipuladas para ellas, se implementa un sistema de notificaciones a los alumnos. Este sistema le permite al estudiante escoger el día en que se le enviará la notificación, el medio de comunicación ha usar, y la ventana de tiempo por la que se notifican las evaluaciones (por ejemplo, las evaluaciones de la próxima semana, o la de las próximas 2 semanas, u otra ventana de tiempo), e incluso el estudiante puede escoger no recibir estas notificaciones.
3. *Mejora en la gestión de fechas de evaluaciones:* El sistema legado notifica a quién corresponda, sobre todos los cambios realizados en las fechas de evaluaciones de aquellos cursos obligatorios. La nueva versión del software monitorea las fechas de todos los cursos de malla, y notifica a los funcionarios del área docente de situaciones que requieran su atención, cuya definición se

deduce de una encuesta realizada a un grupo de estudiantes del departamento. Ejemplos de estas situaciones son las colisiones de fechas de evaluaciones para cursos de un mismo semestre según la malla, o el exceso de evaluaciones dentro de una semana. Además, se implementa un nuevo servicio que permite sugerir fechas para las evaluaciones de cursos ya dictados por el profesor; esto se hace en base a las semanas en que se realizaron previamente dichas evaluaciones.

4. *Sistema de roles y login*: Si bien el sistema legado cuenta con un sistema de usuarios y login con UPasaporte, en esta memoria se abordan problemáticas referentes a la robustez del sistema de roles, como por ejemplo, la mantenibilidad de la conexión del servicio de login con UPasaporte. En esta nueva versión del sistema, se aumenta el número de roles de usuario agregando el tipo de usuario y restringiendo el acceso a acciones e información según cada tipo. Además, conecta el servicio de login del sistema, con el existente en el Portal DCC.

Debido a los cambios para mejorar el sistema, se modifican tanto los servicios de backend como de frontend, los cuales presentan un código “legado” que necesita ser entendido e intervenido al momento de implementar dichos cambios. En este sentido, se enfrenta también el desafío de mantener ciertas funcionalidades del sistema intactas, lo que se podría verificar de manera sencilla con un ambiente de testing. Sin embargo, UCalendar no presenta ningún test automatizado o documentación de su API, lo que supone otro desafío al momento de verificar el correcto funcionamiento del sistema al realizar dichos cambios.

### 1.3 Objetivos de la memoria

El objetivo principal de este trabajo de título es dotar a UCalendar con un conjunto de servicios que le permitan ser puesto en producción, así como también robustecer y mantener aquellos ya existentes, para que de esa manera se facilite la gestión de las evaluaciones de los cursos de la carrera de ingeniería civil en computación. Se espera que esta mejora impacte positivamente las labores de los funcionarios del área de docencia del DCC, y a la organización académica de los estudiantes, permitiendo también aumentar la visibilidad y el control sobre las evaluaciones.

Por su parte, los profesores y auxiliares requieren contar con un sistema de alertas sobre fechas de evaluaciones que ayuden a evitar la sobrecarga académica de los estudiantes dentro de lo posible. Además, el sistema necesita permitir a los docentes, contar con sugerencias sobre las fechas para colocar sus evaluaciones en base a instancias anteriores del curso dictadas por el docente. Y finalmente, los estudiantes deben contar con un servicio de notificaciones que les ayude a planificar el uso del tiempo, haciéndolos conscientes de las evaluaciones que tienen por delante.

Los objetivos específicos que se derivan del objetivo general de UCalendar son los siguientes:

1. Refactorización del código fuente del sistema, para permitir escalar y mantener el proyecto.

2. Habilitar la alimentación de datos desde la base de datos de docencia, hacia la base de datos de UCalendar.
3. Implementar soporte para usuarios estudiantes, para lograr tener un mejor control sobre los servicios que se les entrega; incluyendo las notificaciones y la personalización de la información de los cursos que ellos toman.
4. Mejora de la gestión de fechas de evaluaciones, simplificando el trabajo de Secretaria Docente.
5. Modificar el sistema actual de autenticación, por el “single sign on” implementado por el DCC.

## 1.4 Descripción general de la solución

La solución a desarrollar para abordar los objetivos planteados, consiste en la extensión y el robustecimiento de la plataforma ya existente, con el fin de dejarla disponible en producción a la brevedad posible. Esto permite reemplazar el sistema de gestión de evaluaciones actual, y sirve como herramienta de organización para los alumnos del DCC. Para lograr llegar a este resultado es necesario estudiar el estado de la plataforma legado, analizando su arquitectura, funcionalidades, interfaces y sistema de navegación, permitiendo estudiar los puntos críticos de mejora dentro de la aplicación y poder generar los requerimientos de esta.

Posterior a esto es necesario analizar las posibles soluciones y modificaciones al sistema, donde una de las principales modificaciones a nivel estructural, es realizar su mapa de navegación disminuyendo la cantidad total de vistas significativamente, y simplificando el sistema previo de navegación. Además, de realizar cambios a nivel de arquitectura física y lógica del sistema, con el fin de proceder con la implementación de las funcionalidades necesarias dentro del proyecto.

## 1.5 Estructura del documento

A continuación se muestra cómo se encuentra estructurado el resto del documento de memoria:

- *Capítulo 2. Análisis de UCalendar:* En este capítulo se describe y analiza la plataforma de UCalendar previo a sus modificaciones, incluyendo su estructura, funcionalidad, la navegación del sistema y las principales interfaces de este.
- *Capítulo 3. Reingeniería del Producto:* Dentro de este apartado se presentan los requisitos con los que debe cumplir la solución, y se muestra tanto la estructura como las interfaces diseñadas de la misma.
- *Capítulo 4. Solución Implementada:* En este capítulo se presenta la arquitectura y el modelo de datos bajo el cual se desarrolló la solución.

- *Capítulo 5. Evaluación de la Solución:* En este apartado se describe la evaluación de la solución implementada, incluyendo testing a nivel de APIS, validaciones de usuarios y validación de usabilidad del sistema.
- *Capítulo 6. Conclusiones y Trabajo a Futuro:* Dentro de este capítulo se analiza el trabajo realizado y sus resultados finales, además se analiza el cumplimiento de los objetivos planteados así como el trabajo por venir en siguientes iteraciones del sistema.

## Capítulo 2

# Análisis de UCalendar

Para abordar el objetivo general del proyecto, se analiza el contexto a partir del cual surge la plataforma UCalendar, así como también su estado inicial (sistema legado). Basado en ese conocimiento, se identifica cómo y dónde modificar los distintos servicios de la aplicación, lo cual permite cumplir, tanto con los objetivos de la memoria, como con los requisitos de la nueva versión de la herramienta.

### 2.1 Contextualización del proyecto

Durante el primer semestre del año 2019, el alumnado de la Facultad de Ciencias Físicas y Matemáticas (FCFM) de la Universidad de Chile se movilizó en búsqueda de una mejor calidad de vida y salud mental. A partir de esto se visibiliza la sobrecarga académica del alumnado de todos los Departamentos, incluyendo al Departamento de Ciencias de la Computación (DCC).

Con el ideal de ayudar a mitigar esta carga académica, la Secretaría Docente del DCC buscó mejorar la distribución de evaluaciones (controles y tareas) a través de un procesamiento manual de la información. Como una forma de facilitar esta labor y reducir las limitaciones típicas del procesamiento manual de la información, nace la iniciativa UCalendar.

Como ya se mencionó, UCalendar es una plataforma Web enfocada en mejorar la distribución de la carga de evaluaciones de los alumnos del DCC. Esta aplicación permite visualizar en un calendario o mapa de calor, la carga académica asociada a las distintas semanas del semestre, con el fin de mejorar la distribución de evaluaciones entre los distintos cursos de un mismo semestre de la malla.

Previo a la implementación del proyecto, existían cuatro roles para los usuarios de la aplicación (particularmente: administradores, profesores, auxiliares y ayudantes), los que pueden presentarse de manera simultánea en la aplicación. Existen permisos asociados a estos usuarios, donde el nivel de permisos decrece de administradores a ayudantes, donde los primeros pueden modificar todo el contenido de la aplicación, y los últimos sólo son capaces de visualizarlo. Estos permisos se encuentran distribuidos entre el backend y el frontend de la aplicación, por lo que se puede obtener cierta información no relacionada con el usuario vía API (Application Programming Interface).

Si bien UCalendar cuenta con perfiles de usuario para realizar diversas tareas, en desarrollos previos se postergó el abordar la problemática asociada a la gran carga de trabajo repetitivo requerido para poder utilizar los servicios de esta aplicación. Es por esto que se acotaron las interacciones de este tipo con el sistema, para así mejorar su experiencia de uso. En particular, son los usuarios administradores y profesores quienes deben realizar una ardua labor, creando nuevos semestres y sus cursos asociados en la plataforma. Esto se realiza en el antiguo sistema cada vez que se da inicio a un nuevo semestre académico.

## 2.2 Estructura del sistema legado

Como se aprecia en la Figura 1, la arquitectura física de UCalendar consiste en dos aplicaciones: una de frontend y una de backend. El frontend está implementado en React<sup>1</sup>, y se encarga de conectar la aplicación con sus usuarios finales a través de una interfaz Web.

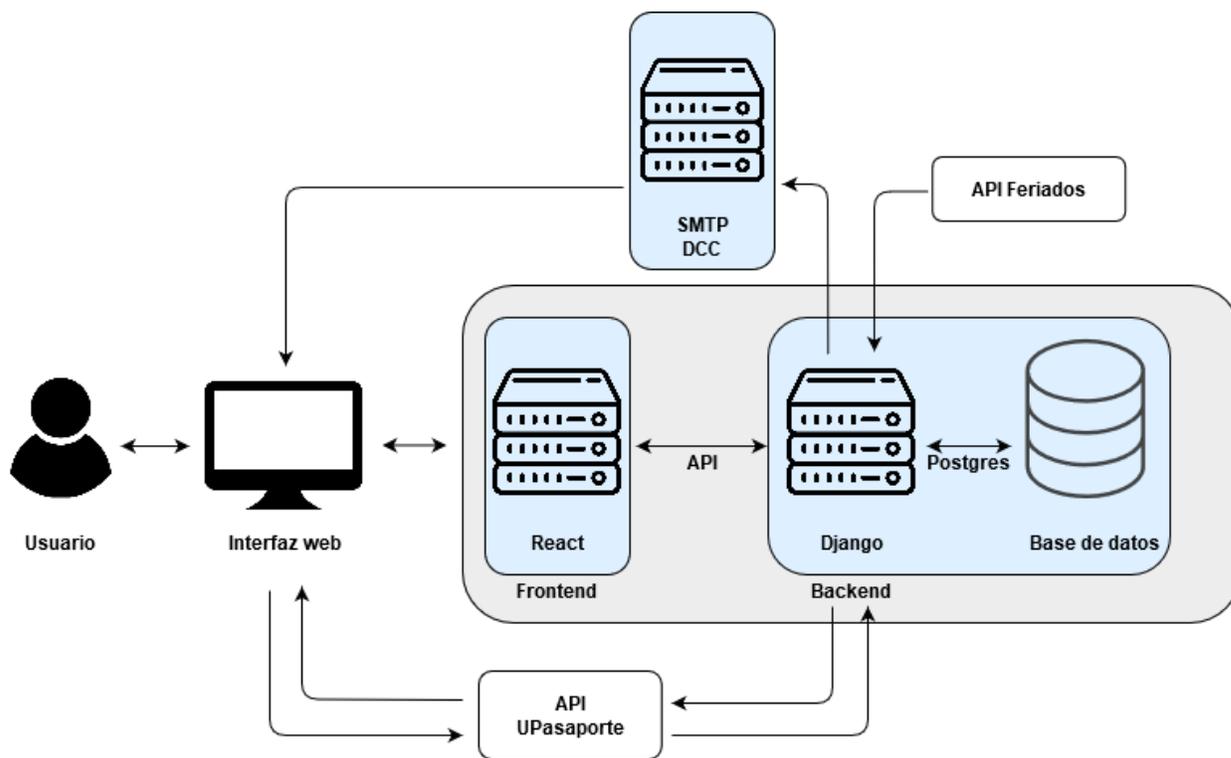


Figura 1. Arquitectura física de UCalendar (obtenido de [1])

El frontend envía y recibe datos del backend de UCalendar a través de su API. Este último se encuentra implementado en Django rest-framework<sup>2</sup>, y se encuentra conectado a una base de datos postgres. Además, el backend de UCalendar tiene conexiones con el servidor SMTP del DCC para el envío de emails, y para autenticar usuarios a través de la API de UPasaporte. En la Figura 1 también se observa una conexión con la API de feriados (la que no se encuentra implementada en el sistema legado).

Como se aprecia en la Figura 2, y tal como se detalla en la sección 2.3 del documento, el usuario de UCalendar interactúa con diversas funcionalidades a través de su interfaz web. Esta interfaz (vista del sistema) se comunica a su vez con el controlador de la aplicación para permitir la ejecución de los servicios.

<sup>1</sup> <https://es.reactjs.org/>

<sup>2</sup> <https://www.djangoproject.com/>

Entre las funcionalidades disponibles podemos encontrar algunas dependencias externas, como lo son la conexión con la API UPasaporte para autenticar a los usuarios, y el envío de notificaciones realizado a través del servidor SMTP del DCC.

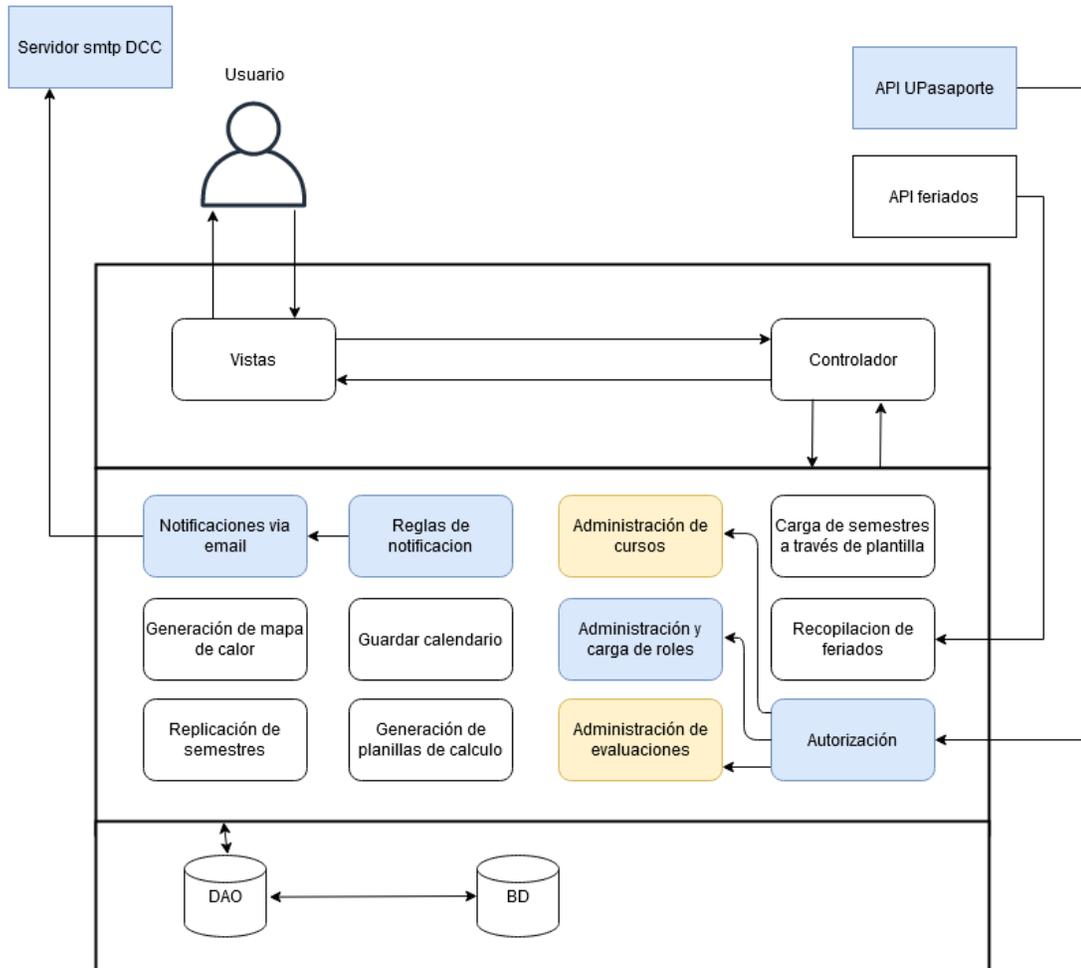


Figura 2. Arquitectura lógica del sistema UCalendar (obtenido de [1])

## 2.3 Funcionalidades del sistema legado

Para cumplir con su propósito, el sistema legado de UCalendar está dotado de diversas funcionalidades, siendo las siguientes las más relevantes:

1. *Administración de Evaluaciones:* Esta funcionalidad permite a los usuarios administradores, profesores y auxiliares administrar (es decir, crear, editar y eliminar) evaluaciones.
2. *Administración de Roles de Usuarios:* El sistema permite administrar roles de usuario (profesor, auxiliar, y ayudante). Esta funcionalidad está disponible para que profesores agreguen auxiliares y ayudantes a sus cursos, así como también para que administradores agreguen nuevos profesores, ayudantes y auxiliares a los diversos cursos.

3. *Administración de Ramos*: El sistema permite administrar (crear, editar y eliminar) ramos sólo a usuarios administradores, asociando a los mismos el semestre al que corresponden según la malla.
4. *Administración de Semestres*: El sistema permite administrar (crear, editar y eliminar) semestres sólo a usuarios administradores.
5. *Administración de Usuarios*: El sistema permite agregar, editar y eliminar usuarios sólo a usuarios administradores.
6. *Clonación de Semestres*: El sistema permite replicar semestres anteriores con igual número de semanas. Con esta funcionalidad se pretende contar con un semestre en el cual basarse, y así simplificar el trabajo repetitivo requerido para la creación de cursos y profesores.
7. *Administración de Cursos*: El sistema permite administrar (crear, editar y eliminar) *cursos*, sólo a usuarios administradores.
8. *Administración de fechas especiales*: El sistema permite administrar *fechas especiales* (es decir, festivos, recesos, y eventos extraordinarios), sólo a usuarios administradores.
9. *Visualización y Almacenar Mapas de Calor*: El sistema permite visualizar un mapa de calor de las evaluaciones distribuidas por fecha, diferenciando los semestres según la malla de la carrera, además de esto permite guardar dicho mapa para ser consultado posteriormente. Cabe señalar que el mapa de calor es generado en base a la carga académica, la cual hasta ese punto es evaluado sólo por la cantidad de evaluaciones presentes en este.
10. *Autenticación de Usuarios*: El sistema permite a los usuarios verificar su identidad para acceder a la plataforma.

## 2.4 Mapa de navegación del sistema

Como se aprecia en la Figura 3, la antigua navegación de la plataforma inicia en el Home de ésta. Si el usuario no se encuentra logueado, entonces es redireccionado a un módulo de inicio de sesión. Una vez iniciada una sesión, el sistema le permite visualizar un semestre y sus respectivos mapas de calor; es decir, el mapa de evaluaciones de los diversos cursos. Desde ahí, a través de una barra de navegación, se podía acceder a diferentes vistas de la aplicación, con el fin de administrar semestres, cursos, evaluaciones, ramos, docentes y *fechas especiales*; en las cuales se está habilitado para hacer operaciones de escritura, lectura, actualización y eliminación (*CRUD*) dependiendo del rol del usuario. A partir de la misma barra de navegación, los usuarios están facultados también para cerrar su sesión, lo que los redirige nuevamente a la ventana login.

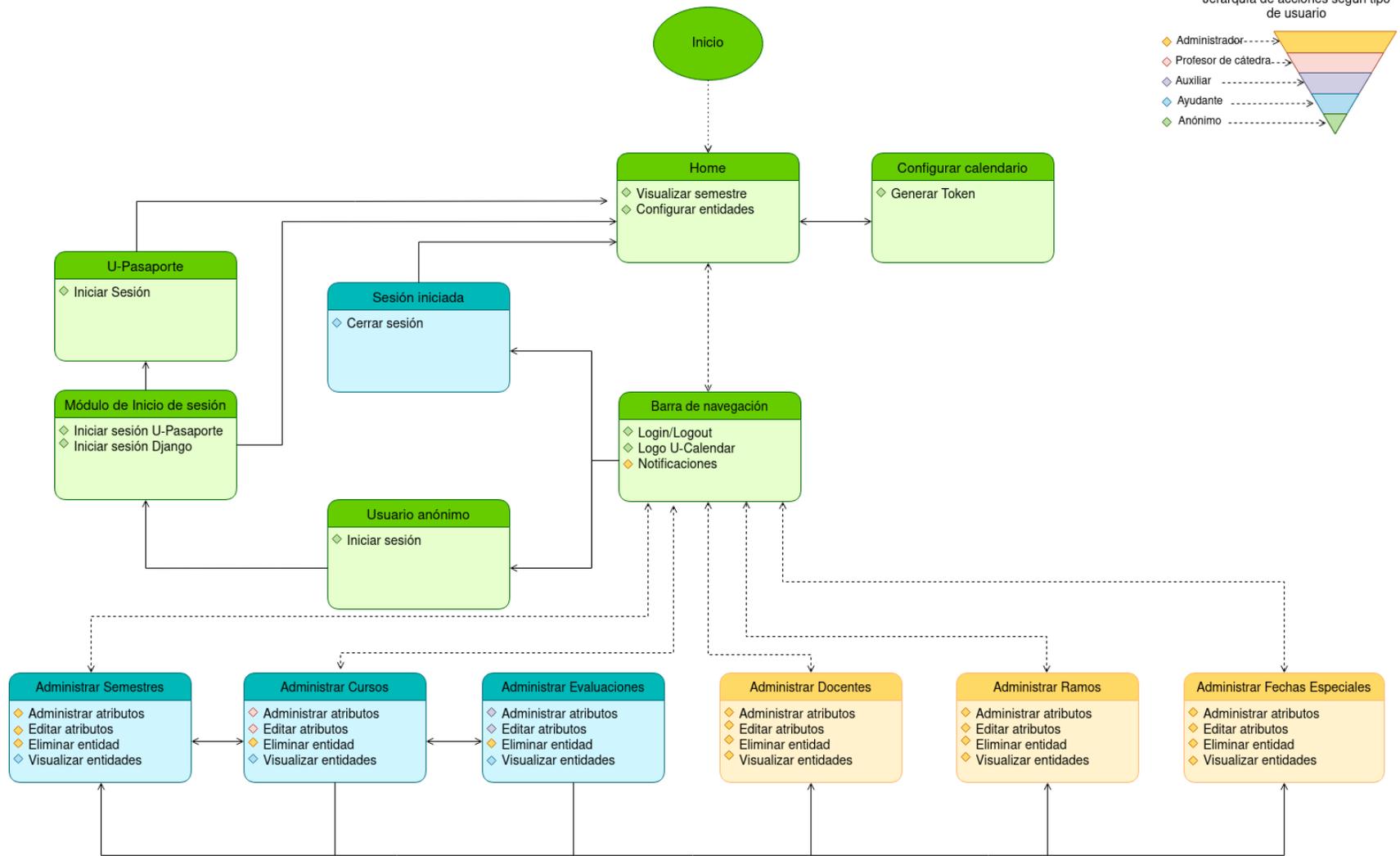
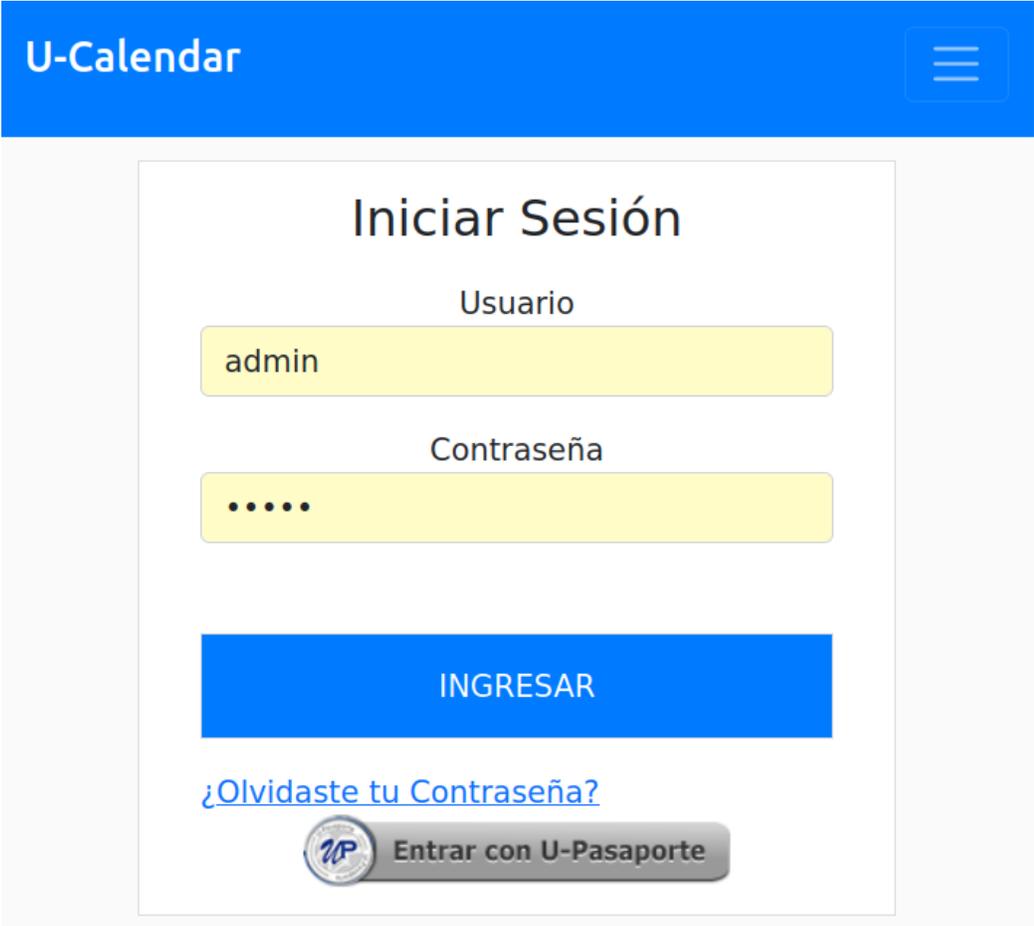


Figura 3. Mapa de navegación de UCalendar (versión base)

Dentro del mapa de navegación podemos observar las acciones realizables por los usuarios dependiendo de su nivel o rol dentro del sistema, destacando que los permisos son acumulables, es decir, un usuario de mayor nivel pueden realizar también las acciones de aquellos de menor nivel. Si bien se observa que se cumple con la “regla de los 3 clicks”, también se puede notar el excesivo nivel de granularidad de sus vistas en donde cada una de estas se encuentra asociada a una sola acción, lo que podría representar una problemática para abordar el crecimiento de la aplicación.

## 2.5 Interfaces de usuario del sistema legado

A continuación se pueden observar las principales interfaces de usuario del sistema, observadas desde el punto de vista de un administrador, y obtenidas del documento histórico del proyecto UCalendar [1]. En la Figura 4 podemos observar el login del sistema, y una redirección a un servicio de login con UPasaporte para la autenticación de usuarios a través de esta vía.



The image shows a web interface for logging into 'U-Calendar'. At the top, there is a blue navigation bar with the text 'U-Calendar' on the left and a hamburger menu icon on the right. The main content area is white and features a login form titled 'Iniciar Sesión'. The form includes two input fields: 'Usuario' containing the text 'admin' and 'Contraseña' containing five dots. Below these fields is a prominent blue button labeled 'INGRESAR'. At the bottom of the form, there is a blue link that reads '¿Olvidaste tu Contraseña?' and a grey button labeled 'Entrar con U-Pasaporte' which includes a circular logo with the letters 'UP'.

Figura 4. Interfaz de login

Por otra parte, en la Figura 5 se muestra el mapa de calor (heatmap) que contiene todas las evaluaciones de los cursos obligatorios de un cierto semestre según la malla. Esta interfaz permite seleccionar el semestre de dictado, y los cursos de los cuales se quiere ver las fechas de las evaluaciones.



### Seleccione Cursos

- Quinto Semestre >
- Sexto Semestre >
- Séptimo Semestre >
- Octavo Semestre >
- Noveno Semestre >
- Décimo Semestre >
- Electivos >

Guardar Calendario

### Heatmap Semestre Otoño 2022

Mes	Sem	Lun	Mar	Mie	Jue	Vie	Sáb	Dom
Mar	S1	28	01	02	03	04	05	06
Mar	S2	07	08	09	10	11	12	13
Mar	S3	14	15	16	17	18	19	20
Mar	S4	21	22	23	24	25	26	27
Abr	S5	28	29	30	31	01	02	03
Abr	S6	04	05	06	07	08	09	10
Abr	S7	11	12	13	14	15	16	17
Abr	S8	18	19	20	21	22	23	24
May	S9	25	26	27	28	29	30	01

### Cursos Seleccionados

- Algoritmos y Estructuras de Datos-1 CC3001

### Leyenda

- 1 Evaluaciones
- 2 Evaluaciones
- 3 Evaluaciones
- 4+Evaluaciones
- 23 Fecha especial

Figura 5. Interfaz de calendario, que incluye el mapa de calor

Además del mapa de calor, en la parte inferior de la interfaz se listan todos los cursos seleccionados por el usuario; es decir, todos aquellos de los cuales se quiere saber las fechas de las evaluaciones.

En la Figura 6 presentan los ramos presentes en el sistema. Allí un usuario administrador puede crear, editar o eliminar ramos, clickeando el boton amarillo y rojo dentro de cada elemento respectivamente. Se aprecia además la existencia de un buscador para facilitar la búsqueda de cursos para un cierto semestre.

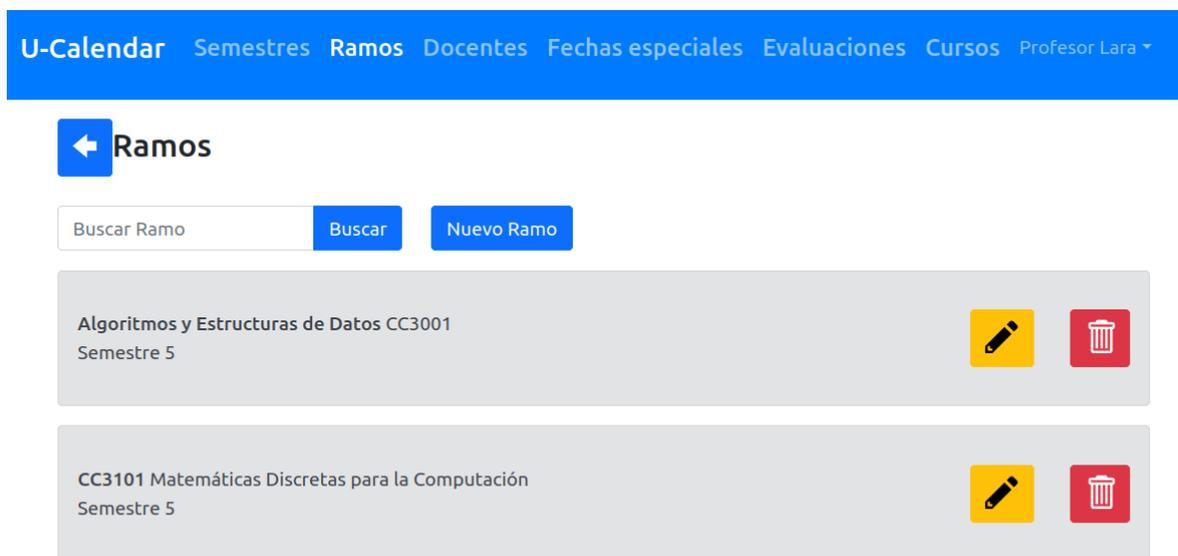


Figura 6. Interfaz de administración de ramos

En la Figura 7 se visualizan las *fechas especiales* definidas por los usuarios administradores para un cierto semestre. Esto corresponde por ejemplo a feriados, semanas de receso, etc. En esta interfaz es posible también crear, editar o eliminar *fechas especiales*.

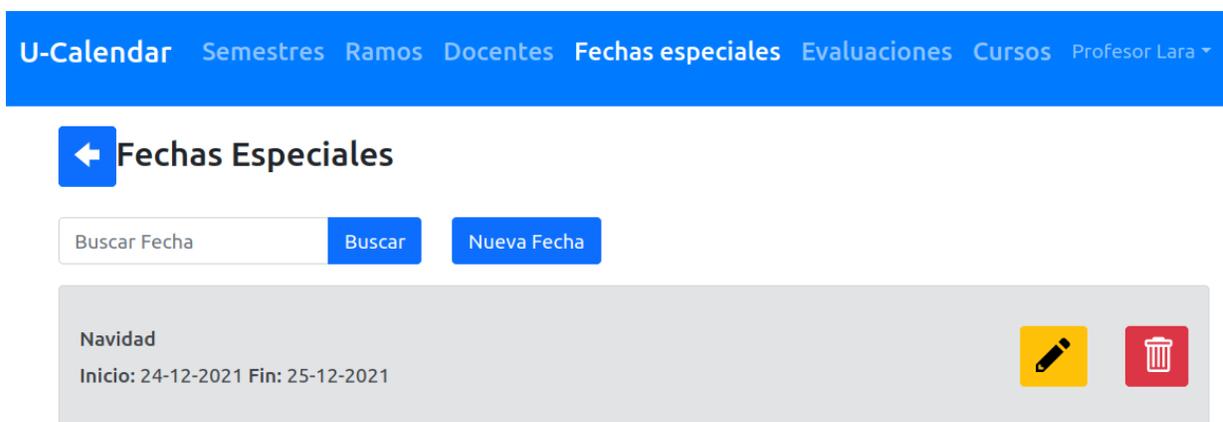


Figura 7. Interfaz de administración de *fechas especiales*

En la Figura 8 se puede observar que, al igual que en las dos figuras anteriores, se pueden realizar operaciones CRUD (Create, Read, Update y Delete) sobre los datos mostrados en la interfaz de usuario, y además buscar elementos. Sumado a esto, desde este punto se puede acceder al mapa de calor utilizando el botón verde dentro de un semestre. También se puede acceder a la vista de cursos utilizando el botón que contiene el libro azul.

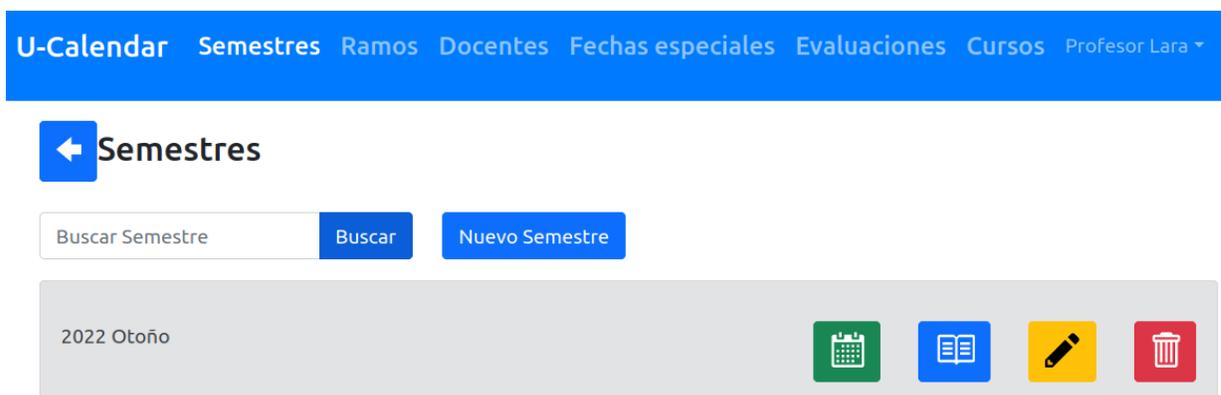


Figura 8. Interfaz de administración de semestres

En la Figura 9 se observa que, al igual que en las tres figuras anteriores, se pueden realizar operaciones CRUD sobre los cursos (esto es, la combinación de ramo y semestre) y de búsqueda de elementos, agregando a esto el acceso a la vista de evaluaciones utilizando el botón azul dentro de cada elemento curso.

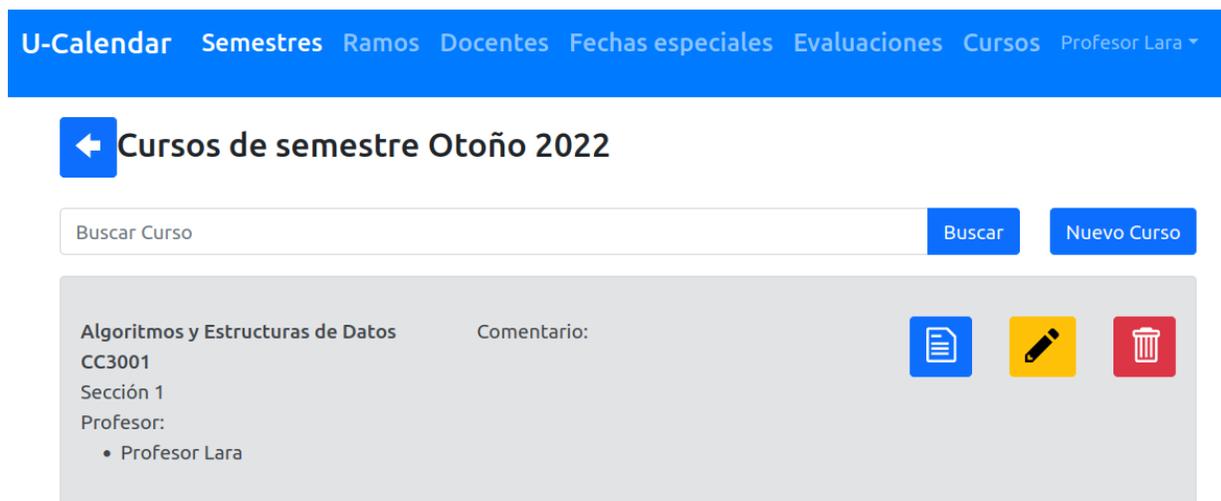


Figura 9. Interfaz de administración de cursos

En la Figura 10 se puede apreciar una tabla con todas las evaluaciones de un curso específico, en donde se visualizan atributos del mismo, tales como su nombre, fecha de inicio y término, su tipo (el cual puede ser tarea o control) y el peso de cada evaluación. Esto último indica el porcentaje de la nota final del curso a la cual corresponde dicha

evaluación; es decir, el peso de la evaluación. En la parte inferior de la interfaz se observa una sección para la creación de evaluaciones, la cual varía según se clickea en el botón de edición o de creación.

## ← Evaluaciones

Curso Algoritmos y Estructu

[Agregar Evaluación](#)

Nombre	Hora Inicio	Fecha Inicio	Hora Fin	Fecha Entrega	Tipo	Peso	
adsaddsa	00:00:00	25-03-2022	10:00:00	25-03-2022	Control	0	 
Ev2	00:00:00	25-03-2022	20:00:00	25-03-2022	Control	10	 
EV1	00:00:00	09-03-2022	00:00:00	14-04-2022	Control	100	 

### Nueva Evaluación

Título\*

Fecha de Inicio\*  Hora Inicio\*

Fecha de Entrega\*  Hora Entrega\*

Tipo\*  Control  Tarea

Peso\*

[Guardar](#) [Cancelar](#)

Figura 10. Interfaz de gestión de evaluaciones de un curso

En general, en las interfaces anteriores se destaca la saturación de colores, en donde los botones presentan una mayor visibilidad dentro de la misma en comparación a la información existente. Esto está motivado tanto por el contraste entre dichos colores, como por su diversidad, lo que puede provocar una desviación en la percepción inicial del usuario al entrar al sistema.

## 2.6 Público objetivo y roles de usuario soportados

El público objetivo de UCalendar está representado en su totalidad por participantes de la comunidad del DCC, en donde se pueden encontrar alumnos, profesores y funcionarios administrativos. En particular, y como se mencionó con anterioridad, dentro de UCalendar se pueden encontrar cinco roles de usuario, los cuales pueden o no presentarse de forma simultánea para un mismo individuo. Estos roles son los siguientes: administradores, profesores, auxiliares, ayudantes y usuarios anónimos.

El sistema actual está diseñado distribuir los roles con respecto al público de la siguiente manera:

- Administrativos: administradores y funcionarios de docencia.
- Profesores: profesores
- Alumnos: auxiliares, ayudantes y usuarios anónimos.

Si bien todos los usuarios pueden estar asociados a uno de estos roles, estos no son del todo restrictivos dentro de la aplicación. Es decir, un usuario anónimo es capaz de acceder vía API a información a la que no debería poseer acceso, como por ejemplo, a los cursos, evaluaciones o semestres a los que no pertenece. Esto podría afectar el correcto funcionamiento del sistema, y la seguridad de la información. Por esta razón, es que además de la modificación y creación de nuevos endpoints en la API para la implementación de nuevas funcionalidades, fue necesario asegurar la calidad del sistema en términos de este aspecto.

## Capítulo 3

# Reingeniería del sistema

Luego de analizar el funcionamiento del sistema legado, así como su código fuente y la estructura del mismo, se definen los requisitos de la nueva aplicación y los puntos de intervención de la misma. Con el fin de alcanzar los objetivos planteados se establecen los principales requisitos para la operación del nuevo sistema, así como las modificaciones a realizar a su estructura, y los diseños relacionados a los cambios de interfaces que responden tanto a una modificación de la navegación, como a cambios en aspectos de usabilidad. Finalmente, se explica el trabajo de refactoring y debugging de código fuente, el cual se realizó para facilitar la extensión de la aplicación de cara a las nuevas funcionalidades que debían implementarse.

### 3.1 Definición de los requisitos del nuevo sistema

Para abordar el trabajo planteado, se definen los siguientes requisitos de usuario, los cuales se derivan de los objetivos de la memoria.

#### **Requisitos de usuario:**

Cada requisito está estructurado mediante un identificador(ID), una descripción, un tipo (funcional, de calidad o de restricción), y perfiles de usuario que hace uso de dicha capacidad del sistema. Además, se indica si el requisito es transversal a todos los perfiles de usuario. En ese caso, se denotará con “-”.

1. Logging UCalendar
  - a. *ID:* RU1
  - b. *Descripción:* Los usuarios deben loguearse en UCalendar usando su cuenta U-Pasaporte, a través del single sign on authentication o Portal DCC.
  - c. *Tipo:* Restricción
  - d. *Usuarios:* -
  
2. FeedBack Estudiantes: Calendario Académico Personalizado
  - a. *ID:* RU2
  - b. *Descripción:* El calendario personalizado filtra los cursos del usuario automáticamente permitiendo visualizar sólo aquellos cursados actualmente como estudiante.
  - c. *Tipo:* Restricción
  - d. *Usuarios:* Estudiantes
  
3. FeedBack Estudiantes: Recepción de Notificaciones
  - a. *ID:* RU3
  - b. *Descripción:* Los estudiantes podrán recibir, a través de mensajería, información sobre su calendario de evaluaciones académicas.
  - c. *Tipo:* Funcional

d. *Usuarios:* Estudiantes

4. FeedBack Estudiantes: Gestión de Notificaciones

a. *ID:* RU4

b. *Descripción:* Los estudiantes podrán configurar la frecuencia, el medio de comunicación y las semanas de antelación con las que se notificará su calendario académico.

c. *Tipo:* Funcional

d. *Usuarios:* Estudiantes

5. Monitoreo de las Fechas de Evaluaciones

a. *ID:* RU5

b. *Descripción:* Los usuarios administradores recibirán notificaciones sobre aquellas fechas de evaluaciones que necesiten una revisión, por ejemplo, debido a la carga académica durante esos días.

c. *Tipo:* Funcional

d. *Usuarios:* Administradores

6. Sugerencias de Evaluaciones de cursos

a. *ID:* RU6

b. *Descripción:* Los usuarios administradores y los docentes recibirán sugerencias de fechas para las evaluaciones.

c. *Tipo:* Funcional

d. *Usuarios:* Administradores, Docentes

7. Funcionalidades según roles de usuario

a. *ID:* RU7

b. *Descripción:* El sistema debe garantizar que las funcionalidades sean accesibles sólo por aquellos usuarios que poseen permisos para ello.

c. *Tipo:* Restricción

d. *Usuarios:* -

8. Alimentación de DB DCC

a. *ID:* RU8

b. *Descripción:* El sistema debe alimentarse desde la base de datos de docencia del DCC.

c. *Tipo:* Restricción

d. *Usuarios:* -

9. Escalabilidad del Sistema

a. *ID:* RU9

b. *Descripción:* El sistema debe ser escalable en el tiempo.

c. *Tipo:* Calidad

d. *Usuarios:* -

10. Usabilidad del Sistema

a. *ID:* RU10

b. *Descripción:* El sistema debe ser predecible y comprensible.

c. *Tipo:* Calidad

d. *Usuarios:* -

### 3.1.1 Alimentación desde la base de datos de docencia

Con el fin de cumplir con el requisito RU8 y alimentar la aplicación desde una base de datos externa, se decide realizar una clonación inicial de la información de datos de semestres, cursos, ramos y sus integrantes, la que está disponible en el data lake del DCC, así como también una constante actualización de dicha información. A su vez, se modifican diferentes funcionalidades de UCalendar, como por ejemplo la administración de semestres, ramos, docentes y cursos, dado que cierta información dependerá estrictamente de la información que se encuentra contenida en el portal, y no debe ser actualizada por el sistema UCalendar. A continuación se expone la información disponible para cada uno de los endpoints a utilizar en la API del datalake del DCC:

1. *Institución:* Endpoint usado para obtener el código de institución asociado al DCC, con el fin de obtener los ramos asociados a éste. Ninguna información de este endpoint será almacenada.
2. *Ramos:* Se usará este vínculo para consultar los ramos asociados al DCC que serán almacenados en la base de datos de UCalendar, obteniendo así su nombre y código.
3. *Periodos:* A partir de aquí se obtendrán los semestres dictados en la FCFM, de los cuales se puede obtener información como su nombre, periodo, año y estado actual.
4. *Cursos:* Asociado a la obtención de instancias de ramos en un determinado periodo de los cursos, se puede consultar información tal como su ramo, periodo, comentario y sección asociada.
5. *Integrantes:* La información de integrantes es usada para identificar a las personas que participan en un determinado curso, como por ejemplo, su cuerpo docente, auxiliares, ayudantes y estudiantes. Esto permite asociar estos tanto a los usuarios de UCalendar como a los a los integrantes de los cursos con su respectivo rol.

La incorporación de este servicio derivó en diversos cambios al sistema UCalendar, los que son dados a conocer a continuación:

1. Modificación de la base de datos
  - a. Derivado de la necesidad de actualizar constantemente los datos obtenidos, es que se almacenan sus identificadores relacionados con el data lake.
2. Cambios en el servicio de API:
  - a. Respecto a Cursos.
    - i. Eliminar el endpoint “agregar cursos”.
    - ii. Impedir la eliminación de cursos.
    - iii. Permitir sólo la edición de los módulos horarios.

- b. Respetto a Semestres.
    - i. Eliminar el endpoint “agregar semestres”.
    - ii. Impedir la eliminación de semestres.
    - iii. Permitir sólo la edición de las fechas límite del semestre (inicio y término del mismo).
  - c. Respetto a Usuarios/Docentes.
    - i. Sólo se permitirá la realización de consultas.
  - d. Respetto a Ramos.
    - i. Eliminar el endpoint “agregar ramos”.
    - ii. Impedir la eliminación de semestres.
    - iii. Permitir sólo la edición del semestre de malla de los ramos.
3. Ajustes a las interfaces de usuario:
- a. Respetto a Cursos
    - i. Modificar los formularios de edición.
    - ii. Eliminar la funcionalidad para agregar o eliminar cursos.
  - b. Respetto a Semestre
    - i. Modificar los formularios de edición.
    - ii. Eliminar la funcionalidad para agregar o eliminar semestres.
  - c. Respetto a Ramos
    - i. Modificar los formularios de edición.
    - ii. Eliminar la funcionalidad para agregar o eliminar ramos.
  - d. Respetto a Usuarios/Docentes
    - i. Eliminar la vista.

### 3.1.2 Monitoreo y sugerencia de evaluaciones

Para el monitoreo de evaluaciones y su respectivo análisis, correspondiente a los requisitos de usuario RU5 y RU6, se requiere identificar el estado de las evaluaciones respectivas. Por ejemplo, si una evaluación representa un estado de alerta, entonces se le asigna un valor correspondiente a éste. Dicho valor se recalcula cada vez que se consulta por la evaluación. Sin embargo, para que este sistema fuese más útil para el beneficiario final, es que se decide realizar una encuesta a alumnos del Departamento de Ciencias de la Computación. Dicha encuesta busca conocer la opinión y percepción del alumnado, respecto al estrés causado por la carga académica en diferentes ámbitos, tanto en aquellos relacionados con la ponderación de las evaluaciones, como también con la cantidad de éstas. De esa manera, se debió realizar un nuevo cálculo de la carga académica, tanto para el mapa de calor como para las notificaciones de recordatorio de evaluaciones que envía el sistema.

Por otra parte, para el análisis de evaluaciones se requiere contar con un distintivo (flag) para separar aquellas evaluaciones que son sugerencias, con respecto de aquellas que se encuentran agendadas en el calendario. Para permitir esto, se agregó el campo “estado” en la tabla “Evaluaciones”, tal como se aprecia en la Figura 11, en donde se destacan las modificaciones de las tablas en amarillo, las eliminadas en rojo y aquellas nuevas en celeste.

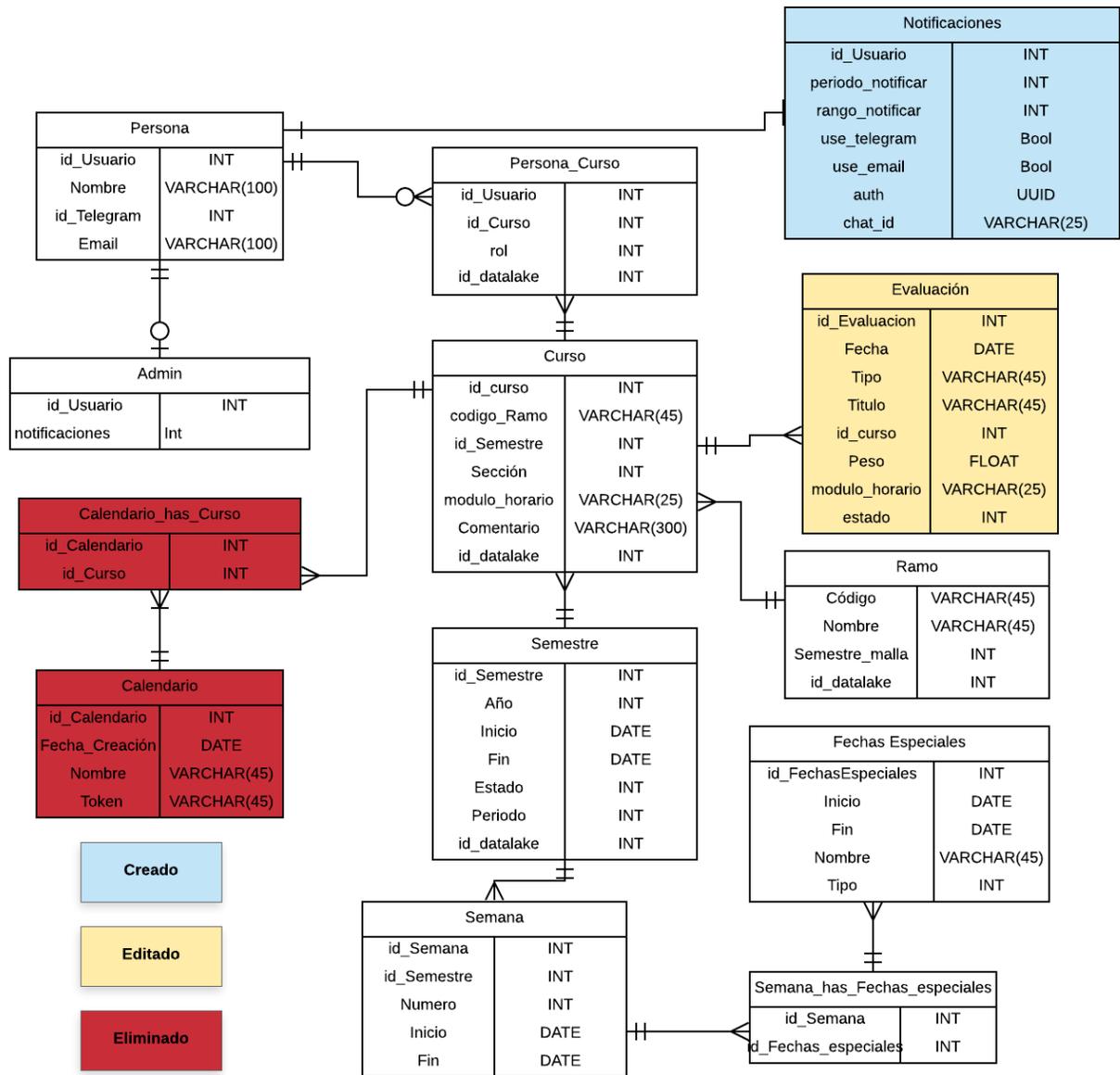


Figura 11. Ajustes de modelo de datos de UCalendar

### 3.1.3 FeedBack a estudiantes: gestión de notificaciones

Para abordar el requisito RU4, y permitir que los estudiantes configuren sus notificaciones para recibir información sobre sus evaluaciones, se agrega la tabla “notificaciones” (Fig. 11). De esta manera, se configura tanto la periodicidad de envío de las notificaciones, como el rango de fechas dentro de las cuales deben estar las evaluaciones a notificar. Así, ningún estudiante se verá abrumado por recordatorios constantes sobre su carga de evaluaciones. Además de esto, se debe crear un modal de configuración para la interfaz, y un servicio en la API que se encarga de manejar estos datos.

De este mismo modo, la gestión de notificaciones depende de un bot de telegram diseñado con el fin de entregar feedback a los alumnos, a través de uno de los medios

de comunicación más importantes dentro de la comunidad de alumnos del departamento. Este servicio es considerado como prioritario al momento de crear un producto mínimo viable del sistema.

### 3.2 Modificaciones a la estructura del sistema

A continuación se explican los cambios en la arquitectura física y lógica del sistema. Tal como se aprecia en la Figura 12, la arquitectura física de UCalendar tiene cambios estructurales, en relación a la Figura 2 (destacados de celeste en la figura 12). El primero es una nueva conexión con la API del data lake de Docencia, la cual permite obtener la información de cursos, secciones, profesores y alumnos para cada semestre, y mantener actualizada la base de datos de UCalendar.

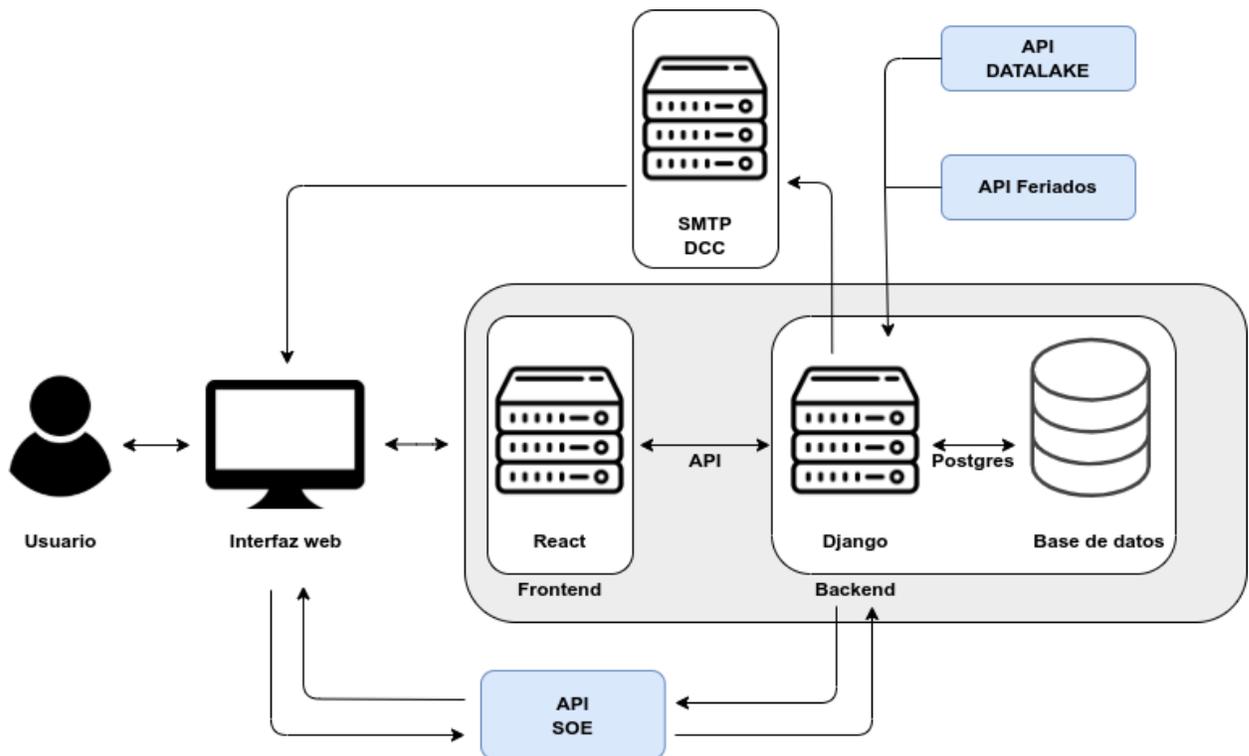


Figura 12. Ajustes de arquitectura física del sistema

El segundo cambio tiene que ver con que se modifica la conexión con la API de UPasaporte para realizar la autenticación de usuarios. Además, se cambia dicha conexión, por un servicio de single-sign-on implementado a través de un endpoint exclusivo del DCC.

En el caso de la arquitectura lógica, tal como se aprecia en la Figura 13, se presentan diversos cambios (destacados de celeste para las nuevas funcionalidades), entre los cuales se encuentra el fin de la clonación de semestres y el almacenamiento de los mapas de calor. Además de esto, se modifica la funcionalidad de la cual depende la autenticación, y se agrega una nueva recopilación de datos de la API perteneciente al data lake del DCC. Por lo demás, existen cambios al sistema de notificaciones, el cual ahora posee dos mecanismos de funcionamiento: uno que se basa en la carga masiva,

y el otro que es sólo para estudiantes. Ambos mecanismos notifican vía email a sus usuarios, cuando corresponda.

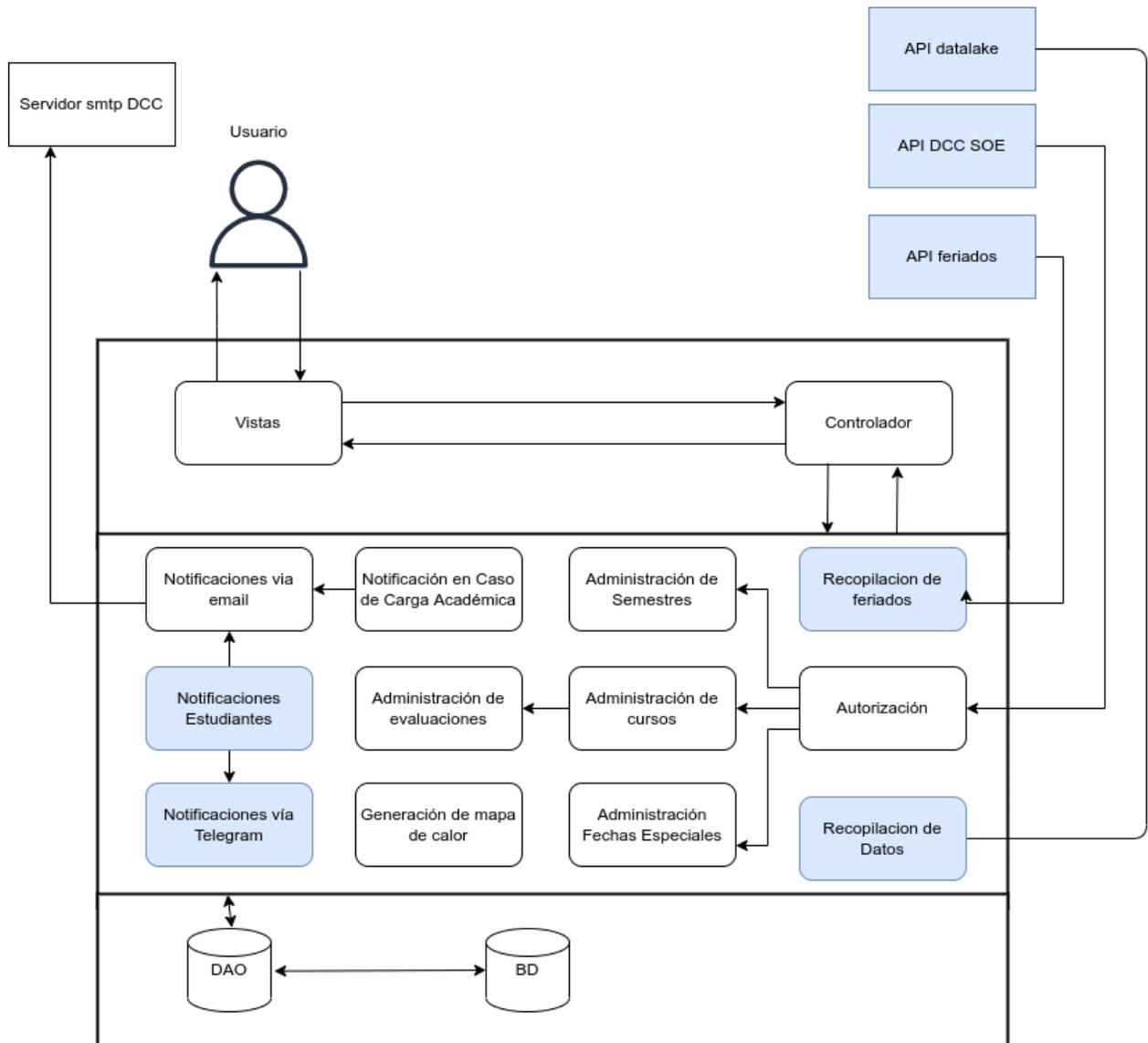


Figura 13. Ajustes a la arquitectura lógica del sistema

### 3.3 Modificaciones a la navegación e interfaces del sistema

#### 3.3.1 Navegación del sistema

Como se aprecia en la Figura 14, el nuevo esquema de navegación de UCalendar inicia en el módulo de Login, si es que el usuario no se encuentra autenticado. En caso de estarlo, éste es redirigido a la vista de calendario, en donde está facultado tanto para visualizar mapas de calor, como para administrar *fechas especiales*. A partir de aquí, el usuario en base a su rol puede dirigirse a diferentes vistas en función de las credenciales que tenga. Por ejemplo, los usuarios de tipo Estudiante y Ayudante

pueden acceder a la vista de Calendario, Semestres y Evaluaciones por curso, utilizando principalmente visualización de datos.

Los usuarios con rol de Auxiliares y Profesores pueden acceder además a la vista de Ramos, y realizar operaciones de edición de evaluaciones (en realidad son operaciones CRUD). Para finalizar, los usuarios Administradores tienen acceso total al sistema, limitando algunas funcionalidades previas como lo son el agregado y la eliminación de cursos, semestres y ramos. Así pues, se simplifica el mapa de navegación anterior, disminuyendo la cantidad de vistas y agrupando diversas funcionalidades relacionadas dentro de una sola vista.

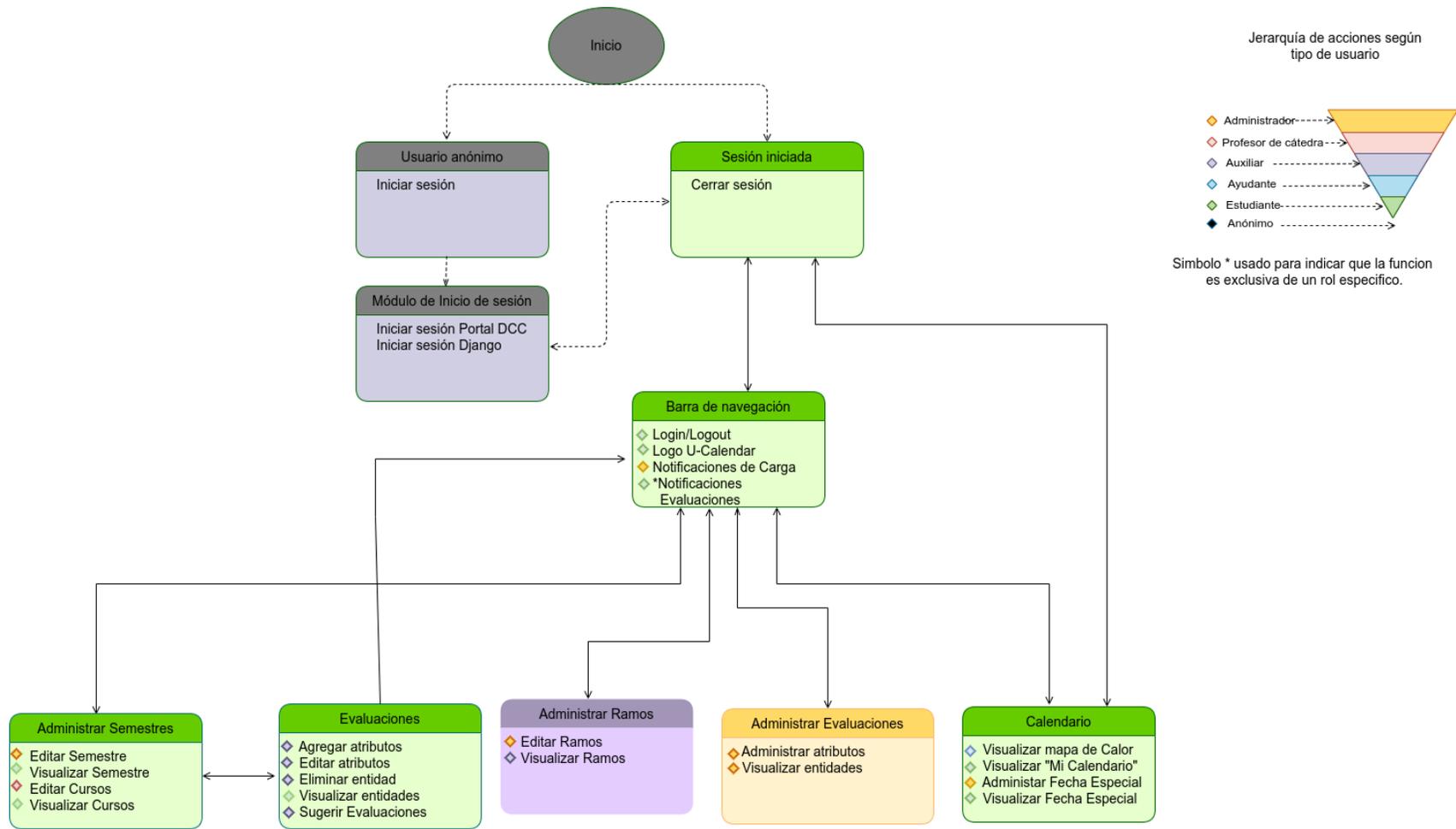


Figura 14. Nuevo mapa de navegación de UCalendar

### 3.3.2 Diseño de las interfaces de usuario

Con el fin de mejorar la experiencia de usuario, e implementar el nuevo esquema de navegación, se cambió el diseño de las interfaces de la aplicación, manteniendo su color azul dado la tendencia de éste a ser asociado con alta calidad en diversas culturas [2]. A su vez, se disminuye el uso de colores con el fin de reducir la saturación. En la Figura 15 se puede observar el diseño para la nueva interfaz de gestión de cursos, que a su vez engloba la antigua interfaz de gestión de semestres.

Esto se implementa mediante un simple sidebar, el cual se encarga de permitir tanto la lectura como la edición de semestres. Además, se listan los cursos mediante cards, las cuales permiten su edición y un acceso a las evaluaciones de las mismas. Cabe señalar que no se presentan formularios de ingreso de estos datos, debido a que es información proveniente exclusivamente del data lake de docencia del DCC.

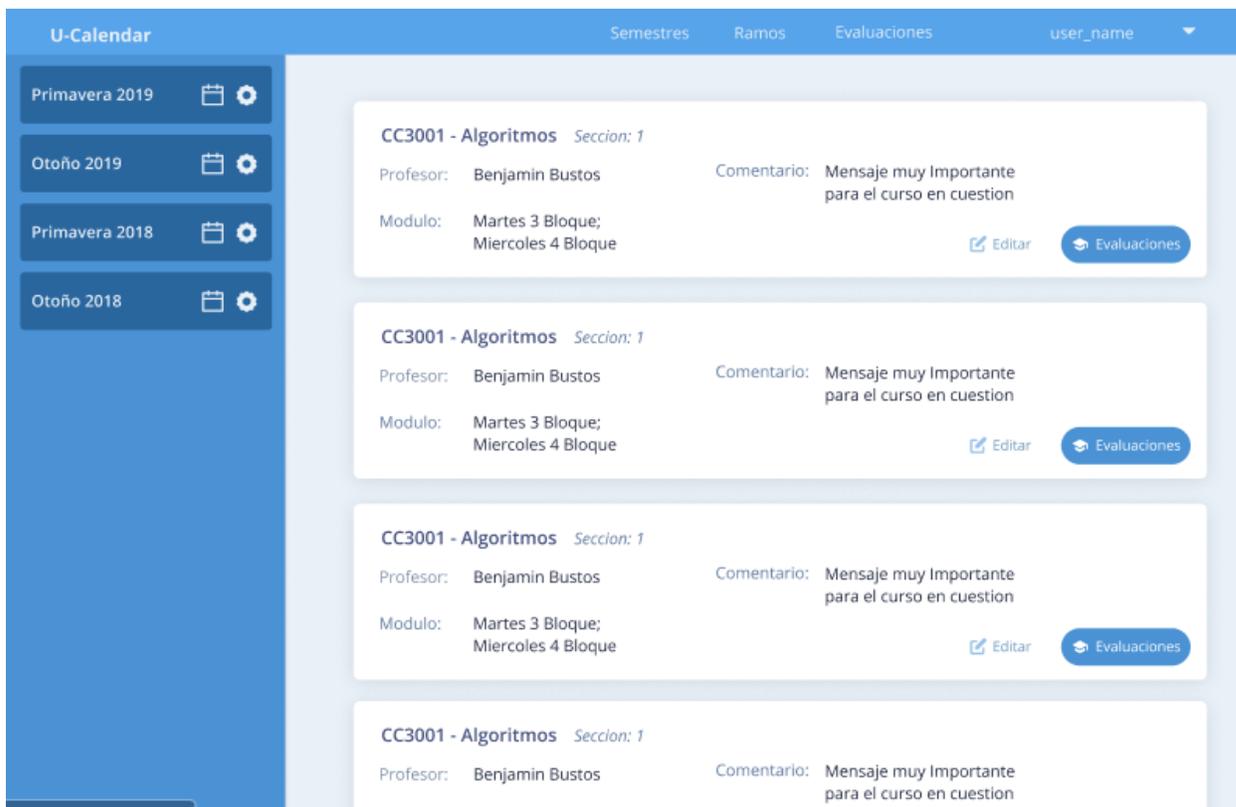


Figura 15. Nuevo diseño de la interfaz de visualización de cursos y semestres

En la Figura 16 se muestra la nueva interfaz de administración de evaluaciones, la cual cuenta con las mismas facultades anteriores, pero con un cambio en su diseño acorde a la interfaz anterior.



Figura 16. Nuevo diseño de la interfaz de búsqueda de evaluaciones

En la Figura 17 se muestra el diseño de la vista para evaluaciones de cursos, donde las secciones inferiores de los formularios se reemplazan por pop-ups. Esto se hace con el fin de disminuir y simplificar la carga visual de la interfaz, y para mantener la consistencia de las interfaces del producto.



Figura 17. Nuevo diseño de la interfaz de evaluaciones de un curso.

En la Figura 18 se puede observar el diseño para la nueva interfaz del mapa de calor. Sobre éste, un usuario administrador puede configurar (operaciones CRUD) las *fechas especiales* del sistema, utilizando el sidebar derecho. Además, bajo el calendario podrá visualizar los eventos asociados a un día particular, por ejemplo, las *fechas especiales* o evaluaciones agendadas, y en el sidebar puede seleccionar el semestre que desea visualizar en el calendario.

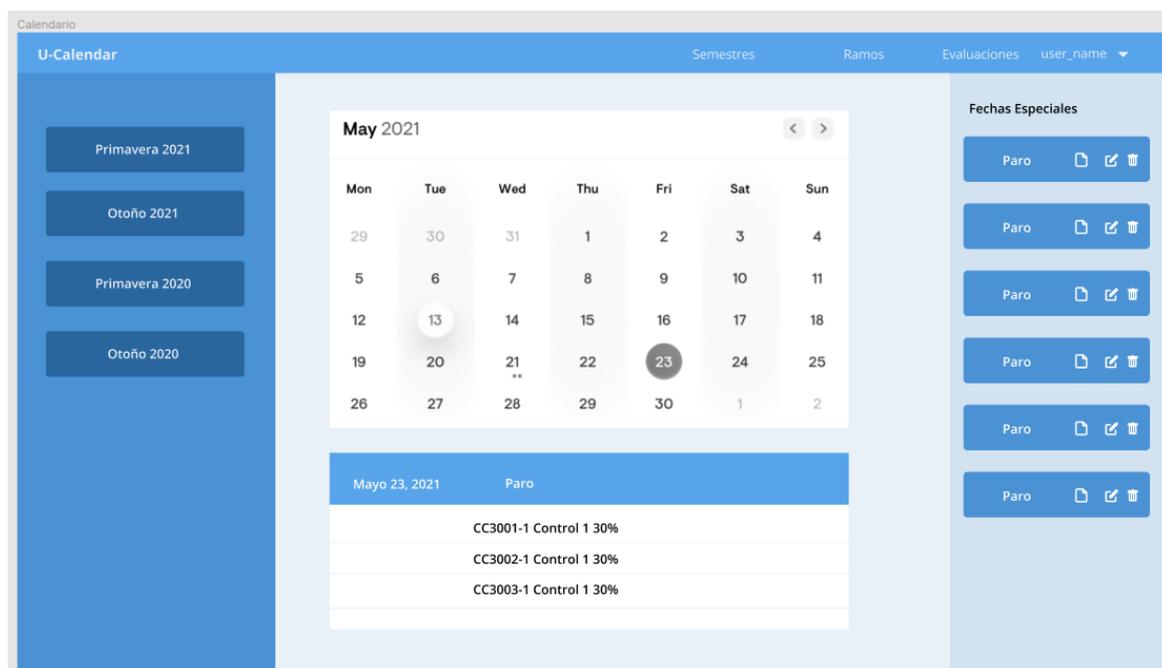


Figura 18. Nuevo diseño de la interfaz del mapa de calor.

En la Figura 19 se observa el diseño para la interfaz de semestres de malla o ramos, la cual es modificada durante el proceso de reingeniería, debido a la imposibilidad momentánea de obtener la información referente al semestre de malla de un ramo en particular. Por lo tanto, se propuso una interfaz que permitía la edición, y visualización según los grupos de semestre de malla correspondientes.



Figura 19. Nuevo diseño de la interfaz de ramos.

### 3.4 Refactoring y debugging del sistema legado

Para facilitar la implementación de nuevas funcionalidades, así como para poder contar con un código mantenible y escalable, se realiza un refactoring del código fuente del sistema. Éste se enfoca principalmente en el frontend de la aplicación, disminuyendo tanto las vistas como sus componentes de navegación. Sin embargo, el backend presenta cambios no menos importantes en su API.

A partir de esto, en el frontend se refactorizan diversos componentes para permitir que estos sean reutilizables dentro de la aplicación. Entre los componentes a refactorizar se encuentran, por ejemplo, aquellos que estaban repetidos en los formularios de edición y creación de elementos del sistema (cursos, evaluaciones, etc.). También se refactorizan elementos de estilo, para facilitar su modificación en el sistema, y se crean componentes asociados al mapa de calor, en base a micro-componentes reutilizables. Esto permite una comprensión más fácil y rápida del código del sistema.

Otro aspecto importante es la reestructuración del código fuente, manteniendo separadas tanto las vistas principales, los componentes generales, los servicios asociados a conexión con API y también aquellos estilos aplicables a lo largo de toda la aplicación.

Por el lado del backend se vieron afectadas principalmente aquellas consultas a los servicios de API de UCalendar, las cuales no consideran los casos de borde, o que no

manejaban ciertos tipos de errores. Por lo tanto, se estandarizó un formato de respuesta para aquellas consultas que retornan algún tipo de error, con el fin de permitir una fácil comprensión para cualquier tipo de usuario.

### 3.5 Herramientas utilizadas

Derivado de los cambios en la aplicación, de las nuevas funcionalidades y de la actualización constante de herramientas, se usaron diversas tecnologías. En el frontend de la aplicación se utiliza Figma como herramienta de diseño de interfaces, ya que permite tratar en forma de componentes a los diversos elementos. Junto con esto, se reemplazaron las herramientas de componentes, por una única herramienta (material UI<sup>3</sup>).

Con el fin de acelerar la velocidad de compilación de este proyecto, y junto con esto la de desarrollo, fue necesario reemplazar webpack por vite. Como cambio final en la aplicación de frontend se actualiza la versión de React utilizada en el sistema, desde su versión basada en clases, a una de componentes basados en funciones y estados.

A modo de resumen, actualmente el frontend de la aplicación utiliza las siguientes tecnologías:

1. React
2. Material UI<sup>4</sup>
3. Redux
4. VITE
5. Figma\*

Para el caso del backend, se eliminó la librería de django que facilita deployar la aplicación en Heroku como se hacía con anterioridad. A su vez, fue necesario dockerizar la aplicación para que su ejecución no dependiera de la máquina en que ésta se montase.

Para el backend del sistema se utilizan tecnologías como la librería “Telegram Bot” que permite una conexión directa con la API de Telegram, de esta manera el sistema puede enviar mensajes a través de esta red; como “CronTab” el cual permite ejecutar trabajos de tipo cron en un servidor, lo que posibilita la ejecución periódica de la actualización en la base de datos. En relación a esta última se utiliza PSQL, ya que se encuentra presente en los ambientes de desarrollo y producción del DCC.

Recapitulando, en el backend se utilizan como principales tecnologías, a las siguientes:

1. Telegram Bot
2. Django Rest Framework
3. Docker
4. Crontab

---

<sup>3</sup> <https://material-ui.com/>

<sup>4</sup> <https://material-ui.com/>

## Capítulo 4

### Solución implementada

En el presente capítulo se muestran los principales cambios que permiten cumplir con los objetivos del presente trabajo. La estructura de las subsecciones del capítulo es la siguiente: 1) modificaciones a las interfaces de usuario, 2) descripción del servicio de alimentación de datos, 3) descripción del servicio de notificaciones a estudiantes, 4) mejora de la gestión de evaluaciones, y 5) soporte para la autenticación de usuarios e implementación de roles.

#### 4.1 Modificaciones de interfaces de usuario

En base al objetivo de simplificar la usabilidad de la aplicación, se procede a rehacer las interfaces del sistema utilizando como referencia los diseños de interfaces presentados previamente en el Capítulo 3.

##### 4.1.1 Vista de calendario

En particular, en esta vista (Figura 20) se aprecia el nuevo calendario de carga académica, cuyo principal cambio es el cómo se mide el nivel de ésta. Esto se explicará en profundidad en el punto 4.4. Por otro lado, existen cambios en cómo se muestran las semanas del semestre, disminuyendo la información y enfocando la atención del usuario en las *fechas especiales* y en las fechas con alta carga académica. Como se ve en la imagen, éstas pueden estar destacadas con diversos colores, siendo aquellos más cálidos los que representan una mayor carga académica para los estudiantes.

Además de las funcionalidades que se aprecian en la Figura 20, existen otras derivadas de los roles de usuario y que buscan adecuarse a las diferentes necesidades según el rol. Dentro de éstas podemos encontrar a las siguientes:

1. Un switch que permite al usuario cambiar del tipo de mapa de calor. Así los usuarios de tipo *estudiantes* que posean un rol de *docencia* pueden visualizar tanto su calendario estudiantil como global.
2. La información asociada a la fecha seleccionada del calendario, mostrando sus *fechas especiales* y evaluaciones.

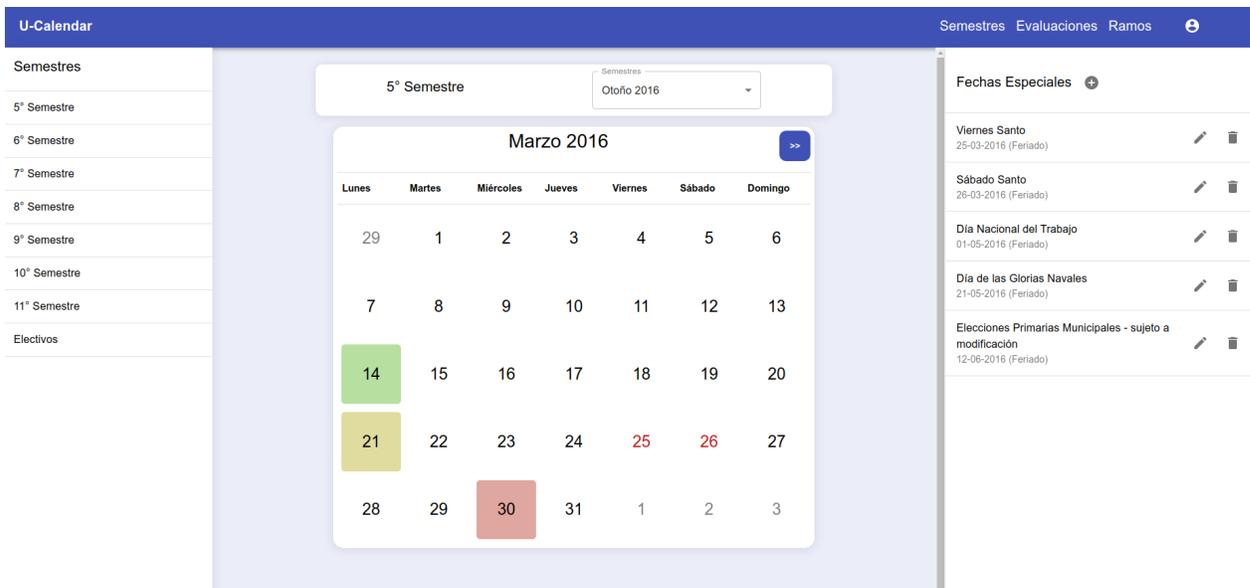


Figura 20. Implementación de la interfaz de calendario.

#### 4.1.2 Vista de semestres

Para el diseño de la vista de semestres, y tal como se aprecia en la figura 21, se realizan cambios derivados de pruebas previas a la evaluación final. De esta prueba se desprenden cambios tales como:

1. La modificación del color del engranaje de configuración de semestres, con el fin de mostrar las inconsistencias a los administradores.
2. Un indicador (banderín) para dar a conocer cuál de los semestres mostrados en el sistema es el actual.
3. Eliminación del vínculo directo al calendario académico, debido a que complejiza la navegación agregando rutas innecesarias al sistema.

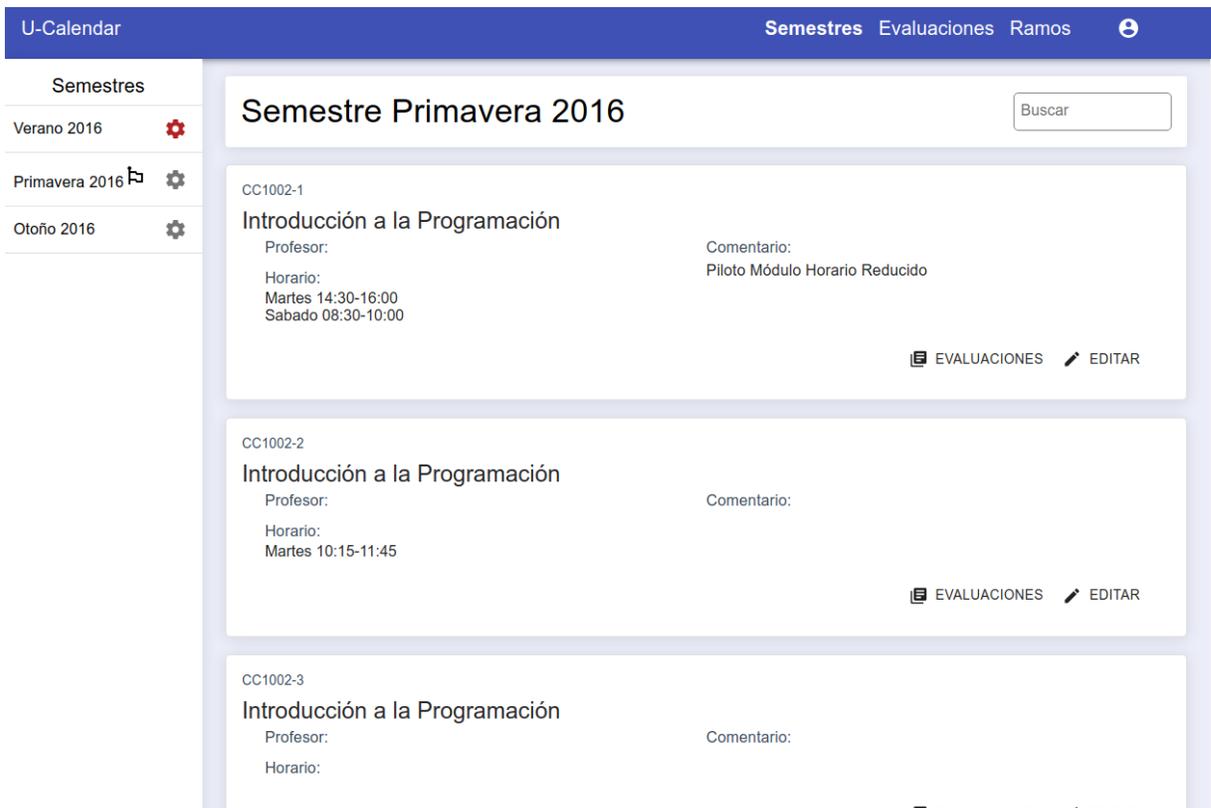


Figura 21. Implementación de la interfaz de semestres.

#### 4.1.3 Vista de evaluaciones

La vista de evaluaciones para cursos (Figura 22) también sufre modificaciones con respecto a su diseño inicial. Esto debido principalmente a la implementación de las sugerencias para evaluaciones. Dado esto, las modificaciones del diseño derivan de:

1. Ejecutar proceso de generación de sugerencias para las evaluaciones futuras, en función de las instancias previas (es decir, evaluaciones pasadas).
2. Operaciones CRUD para manejar sugerencias de fechas de evaluaciones.
3. Funcionalidad para transformar una sugerencia de fecha de evaluación, en una evaluación confirmada.

U-Calendar Semestres Evaluaciones Ramos

**CC3002-1 Metodologías de Diseño y Programación**

< VER SEMESTRES
SUGERIR EVALUACIONES
+ AGREGAR

Nombre	Tipo	Fecha Inicial	Hora Inicial	Fecha Final	Hora Final	Ponderación%		
Evaluacion 1	tarea	09-09-2016	14:11	16-09-2016	01:22	12		

Filas por página: 10 1-1 de 1 < >

**Sugerencias**

Nombre	Tipo	Fecha Inicial	Hora Inicial	Fecha Final	Hora Final	Ponderación%			
Evaluacion 2	tarea	19-10-2016	13:00	24-10-2016	13:00	20	✓		
Evaluacion 3	tarea	14-10-2016	13:00	03-12-2016	16:02	20	✓		

Filas por página: 10 1-2 de 2 < >

Figura 22. Implementación de la interfaz de evaluaciones por curso.

#### 4.1.4 Vista de evaluaciones 2

En la Figura 23 se muestra la interfaz de administración, la cual permite realizar operaciones CRUD sobre las evaluaciones, sin la necesidad de enfocarse en los cursos particulares asociados a éstas.

Al comparar la interfaz con su diseño anterior, el único cambio importante realizado es el de agregar un filtro de *alertas*. Este filtro permite destacar aquellas evaluaciones que caen en fechas problemáticas, ya sea el exceso de carga académica o choques con *fechas especiales*. Este cambio nace debido a que el uso de esta vista se centra principalmente en la modificación y corrección de evaluaciones.

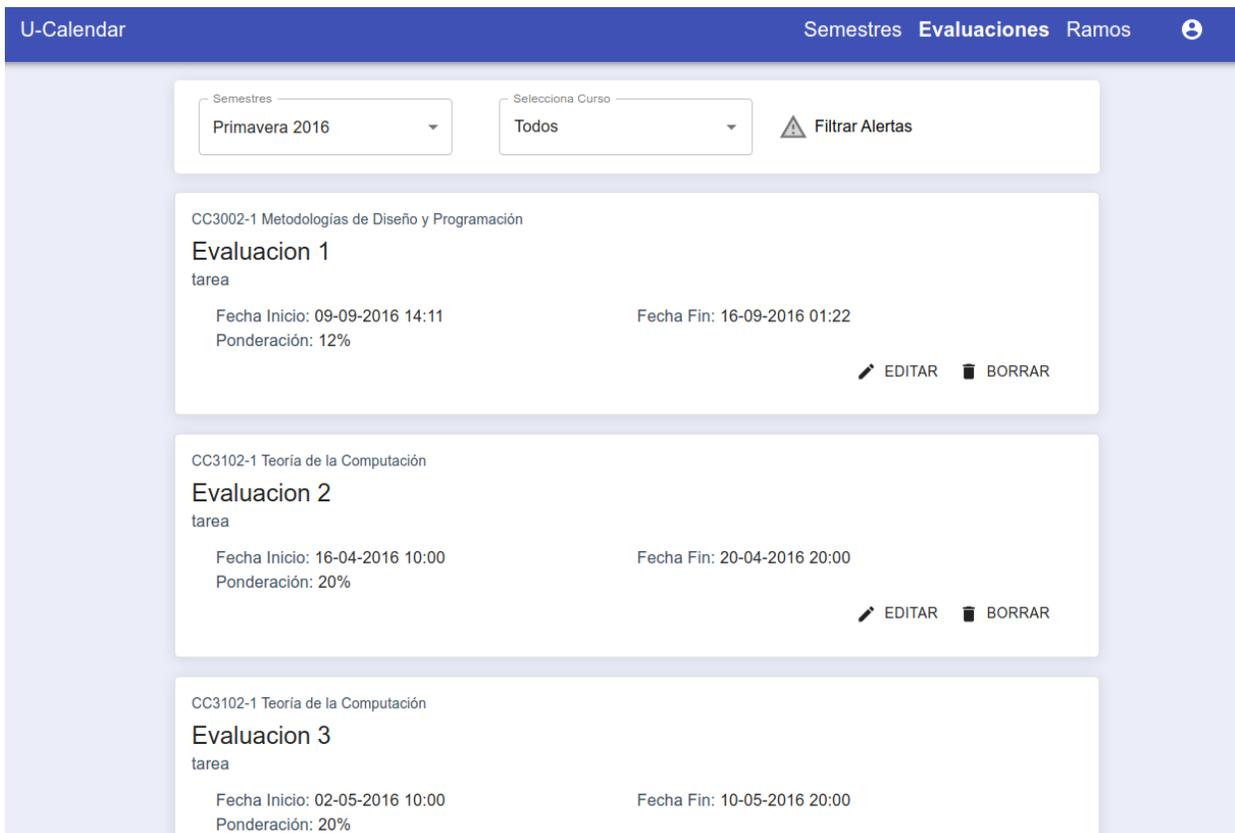


Figura 23. Implementación de la interfaz de administración de evaluaciones.

#### 4.1.5 Vista de ramos

A partir del diseño inicial de la Interfaz de Ramos, se implementan ciertas modificaciones derivadas de los cambios en el flujo de la aplicación, y de la necesidad de re-implementar la edición de ramos. Esto deriva en un componente cuya finalidad radica en informar y editar los ramos asociados a un semestre de malla.

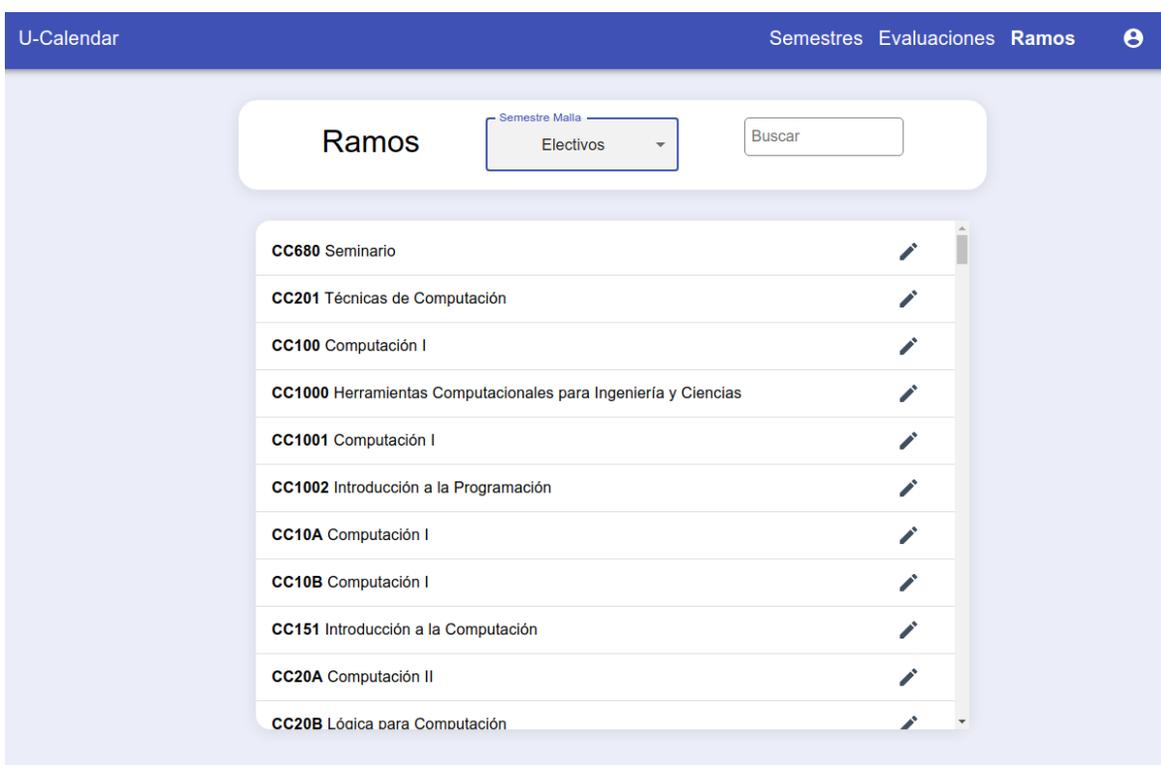


Figura 24. Implementación de la interfaz de ramos.

## 4.2 Servicio de alimentación de datos

Para conectarse al servicio de alimentación de UCalendar, y así obtener los diferentes datos pertinentes desde el Data Lake del DCC, se requieren permisos particulares, los cuales fueron solicitados al área de desarrollo del departamento. Estos permisos permiten conectarse a la API, para la obtención de datos almacenados en la base de datos antes mencionada.

Una vez disponible la autenticación de la API, se procede a identificar los endpoints que contienen la información a ser extraída (clonada) por el sistema UCalendar. Se verifica que estos incluyan la información descrita en su correspondiente documentación, entendiendo que aquella información sería la requerida por la aplicación.

En conjunto a lo anterior, y con el fin de obtener aquellas *fechas especiales* dentro del calendario académico almacenado en UCalendar, se identifica una API gubernamental perteneciente al estado de Chile. Ésta disponibiliza todos los feriados existentes en el país, ya sea por mes y por año. Esta información se puede obtener tan solo desde el año 2012, lo que a futuro puede desencadenar en una limitación si es necesario obtener información retroactiva para la operación del sistema.

Con el fin de mantener la información consistente con la existente en el data lake, y asegurar la actualización constante de la información en el sistema, se crean procesos *cron* dentro del servidor. Estos se ejecutarán periódicamente para consultar por la información existente, y también para identificar posibles cambios en la información de

la API. En particular, los procesos se subdividieron en procesos diarios, semanales y mensuales, los cuales se estructuraron como se muestra a continuación:

1. Diario:
  - a. Integrantes de los cursos
  - b. Cursos
2. Semanal
  - a. Feriados
  - b. Ramos
  - c. Semestres
3. Mensual
  - a. Feriados

Para realizar la subdivisión anterior, es necesario considerar los factores de actualización del data lake (el cual se realiza cada 24 horas), y la periodicidad con la que se realizan cambios en los modelos anteriores, siendo aquellos datos más críticos los que se actualizan de forma diaria en el sistema. Así podemos notar actualizaciones de feriados, tanto mensuales como semanales, donde una se encarga de realizar consultas sobre feriados del mes, y la otra de todo el año. De esa manera, cuando se agreguen o modifiquen feriados con poco tiempo de anticipación, UCalendar se mantendrá actualizado.

### 4.3 Servicio de envío de notificaciones a estudiantes

Para mantener una interacción continua con los estudiantes, se implementa un servicio de notificaciones de las evaluaciones futuras. Este servicio le permite a los estudiantes configurar qué día de la semana quiere que le llegue el recordatorio, y hasta cuántas semanas de evaluaciones futuras (ventana de tiempo) quiere que se le notifique. Así el estudiante se mantiene informado de sus próximas evaluaciones y libre de spam.

Para hacer que esta capacidad del sistema fuese configurable por cada alumno, fue necesario modificar y extender el modelo de datos, con el fin de permitir a cada usuario almacenar su información asociada a las notificaciones. De este modo, se almacena información de contacto, intervalos de notificación, e identificadores de conexión a las distintas plataformas. Con el objetivo de alcanzar el mayor número de estudiantes posibles, las notificaciones se envían tanto por correo electrónico como a través de mensajería de telegram.

Para realizar el envío de mensajes utilizando telegram, se requirió crear un bot utilizando la herramienta “Bot Father”, la cual permite crear y conectarse a un bot utilizando la librería de telegram y una key. Este bot posee dos funcionalidades; además del envío de notificaciones, éstas permiten conectar a los usuarios con su cuenta de UCalendar. A continuación se detallan brevemente estas funcionalidades:

1. *Vincular Cuenta*: Se vincula la cuenta de telegram con su respectiva cuenta en UCalendar, mediante el uso de un UUID4. Esto se hace con el fin de evitar la suplantación de cuentas, almacenando en el sistema el id del chat conectado.

2. *Desvincular Cuenta*: Elimina los datos de notificación en telegram de la cuenta que active esta funcionalidad, evitando así que se le envíen nuevas notificaciones.

## 4.4 Mejora de la gestión de evaluaciones

Con el fin de mejorar el actual sistema de gestión de evaluaciones se realizaron varios cambios al sistema, dentro de los cuales podemos encontrar los siguientes:

1. Sistema de notificaciones de alerta.
2. Sugerencia de fechas de evaluaciones.
3. Modificaciones al mapa de calor

### 4.4.1 Sistema de notificaciones de alerta

Cuando una fecha con algún tipo de problema en alguna evaluación sea detectada dentro del semestre, se envía una notificación a los usuarios administradores. De esta manera, los usuarios del área de docencia ya no deben abocarse a revisar todas las evaluaciones que sufren cambios en el sistema, sino solo aquellos casos que el sistema identifica como problemáticos. Esto reduce significativamente el tiempo de monitoreo de fechas de evaluaciones, y hace a este proceso menos dependiente de las personas.

### 4.4.2 Sugerencia de evaluaciones

Para ayudar a los profesores que siguen una cierta estructura de evaluaciones en sus cursos semestre a semestre, el sistema genera sugerencias de fechas para las evaluaciones en función de información de semestres pasados. Si bien se abordan otros métodos de sugerencias, utilizando las fechas disponibles, resulta complejo prever cuál de estas resultaría una mejor opción dado los ramos con los que podría presentar inconvenientes. Es por eso que una solución a largo plazo, como la mencionada en un principio, se ajusta mejor a las necesidades actuales, aunque ésta presente debilidades al momento de generar las primeras instancias de los cursos, y sea poco aplicable en aquellos no estructurados.

### 4.4.3 Posponer el calendario académico

La solución que está relacionada con posponer el calendario académico en caso de un evento especial, como lo es por ejemplo una paralización estudiantil (que llevará a postergar el término del semestre), presenta la necesidad de ser descartada por el momento debido a la complejidad de abordarla apropiadamente en el marco de esta memoria.

Una incorrecta implementación de esta funcionalidad podría generar una alta carga de trabajo asociado al reajuste del semestre, a partir de una base desconocida para los usuarios. Por lo tanto, se decidió no abordarla mientras no se cuente con un plan establecido por la FCFM para lidiar con este tipo de instancias.

#### 4.4.4 Modificación del mapa de calor

El nuevo mapa de calor permite visualizar aquellos conflictos en las evaluaciones, ya sea con *fechas especiales* u otras evaluaciones (tal y como se hacía anteriormente). Este mapa permite además a los usuarios Estudiantes visualizar su calendario académico, es decir, mostrar sólo aquellos cursos que están siendo cursados por ellos en el semestre actual. De esta manera, se permite a los usuarios organizar sus semanas más complejas con un tiempo de antelación prudente.

Para obtener aquellas situaciones de carga académica relevantes para los estudiantes, fue necesario realizar una encuesta breve para conocer su opinión. La encuesta fue respondida por 80 estudiantes del DCC y se intentó obtener una mirada sobre tres puntos de vista (estos puntos se explican a continuación):

1. Importancia de la cantidad de evaluaciones para un mismo día.
2. Relevancia de las ponderaciones de las evaluaciones de forma diaria.
3. Carga de las ponderaciones de evaluaciones en instancias semanales.

##### 4.4.4.1 Percepción de cantidad de evaluaciones para un día.

Esta instancia plantea 2 casos y se requiere conocer cuál de estos representa un mayor estrés para los alumnos. El “Caso 1” representa un día con 3 evaluaciones, las cuales valen un 20% de la nota final cada una. El “Caso 2” corresponde a un día, con una evaluación cuyo porcentaje de la nota final es de un 60%. En ambos casos las ponderaciones totales de las evaluaciones son idénticas, pero varía la cantidad de evaluaciones para la fecha.

En el gráfico de la figura 25 se puede observar que un 55% de los estudiantes consideran una mayor carga académica el Caso 1. Es decir, consideran que un número mayor de evaluaciones les afecta más negativamente. Un 26,2% de los alumnos opina lo opuesto, y un 18,8 % considera que no es un factor relevante.

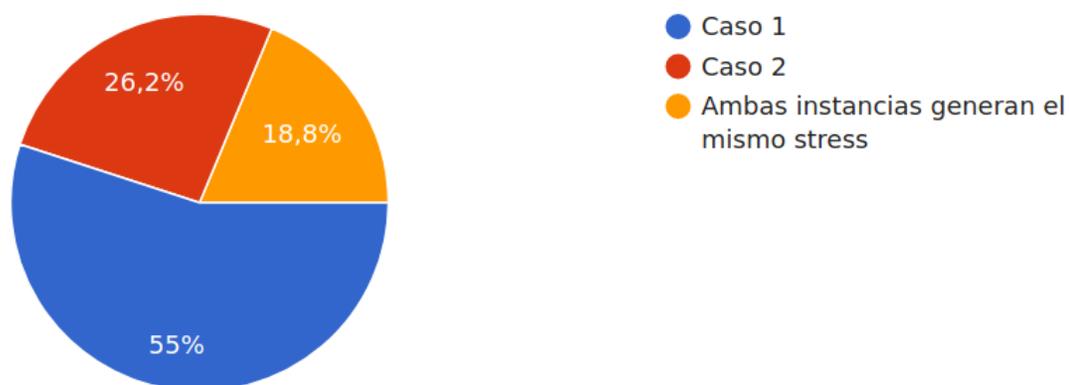


Figura 25. Preferencia de la cantidad de evaluaciones sin variar la ponderación.

#### 4.4.4.2 Percepción de carga aceptable para un día

A continuación se presenta una instancia en donde se consulta a los alumnos qué porcentaje de la nota de evaluaciones les hace pasar cierto nivel de estrés. Los niveles de estrés están definidos como mínimo, medio y alto; todo esto considerando que las evaluaciones se realizan sólo en un día de la semana.

En la figura 26 se aprecian los resultados de la encuesta referente a qué porcentaje de la ponderación total de evaluaciones le causa un estrés bajo a un estudiante. Este valor de ponderación es en promedio un 19.95%, y tiene una mediana de 20% para evaluaciones dentro de un mismo día.

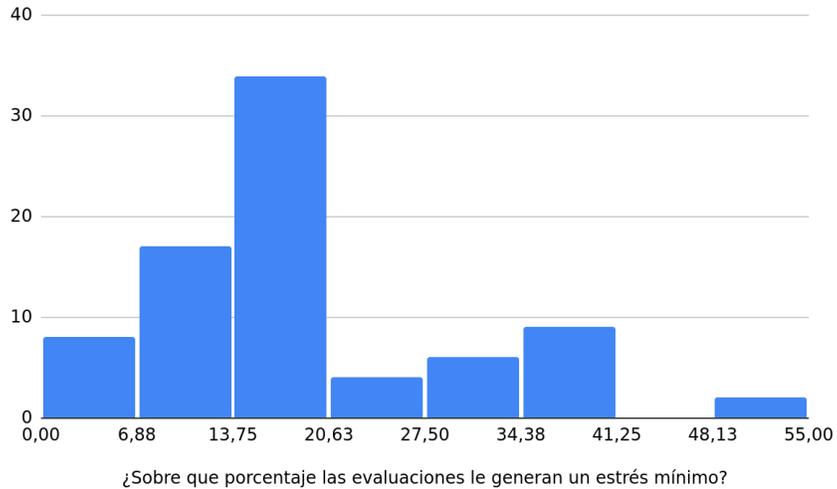


Figura 26. Ponderación total que causa estrés mínimo en un día.

Observando la figura 27 se notan los resultados de la encuesta relacionados con el porcentaje de la ponderación total de la nota de evaluación, que le causa un estrés medio a los estudiantes. En este caso, el promedio es una ponderación del 35.16 %, y la mediana es de un 34 % para evaluaciones dentro de un mismo día.

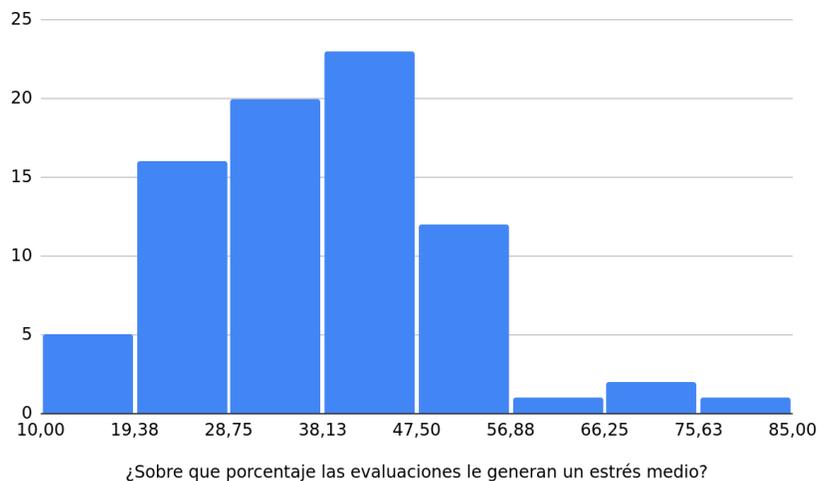


Figura 27. Ponderación total que causa estrés medio en un día.

En la figura 28 se observan los resultados de la encuesta referentes a qué porcentaje de la ponderación total de evaluaciones le causa un alto estrés a un alumno. En este caso, la ponderación promedio es de un 52.25%, y la mediana es de un 50 % para evaluaciones dentro de un mismo día.

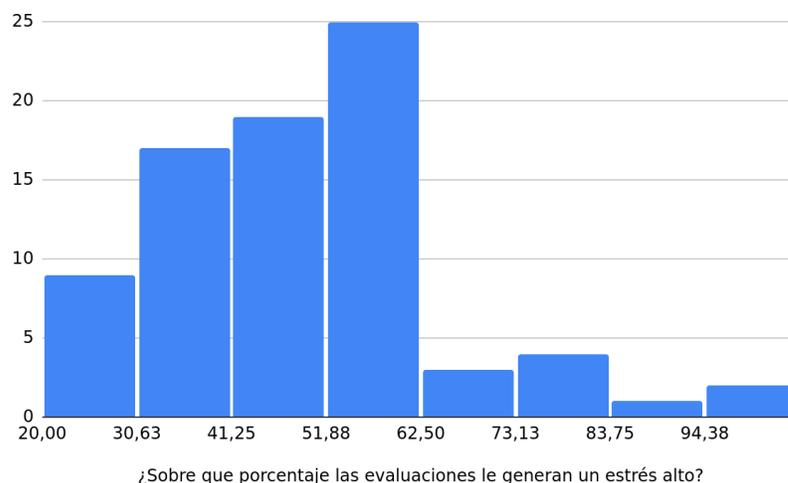


Figura 28. Ponderación total que causa alto estrés en un día.

#### 4.4.4.3 Percepción de carga aceptable para una semana

Para este tercer punto se pregunta al estudiantado qué porcentaje de la nota final, que representan las evaluaciones, les hace pasar cierto nivel de estrés, el cual puede ser mínimo, medio y alto, donde las evaluaciones se realizan a lo largo de toda la semana.

En la figura 29 se aprecian los resultados de la encuesta en relación al porcentaje de la ponderación total de evaluaciones que le causa un estrés bajo a un estudiante, siendo en promedio un 23,18% y la mediana de 20% para evaluaciones dentro de una misma semana.

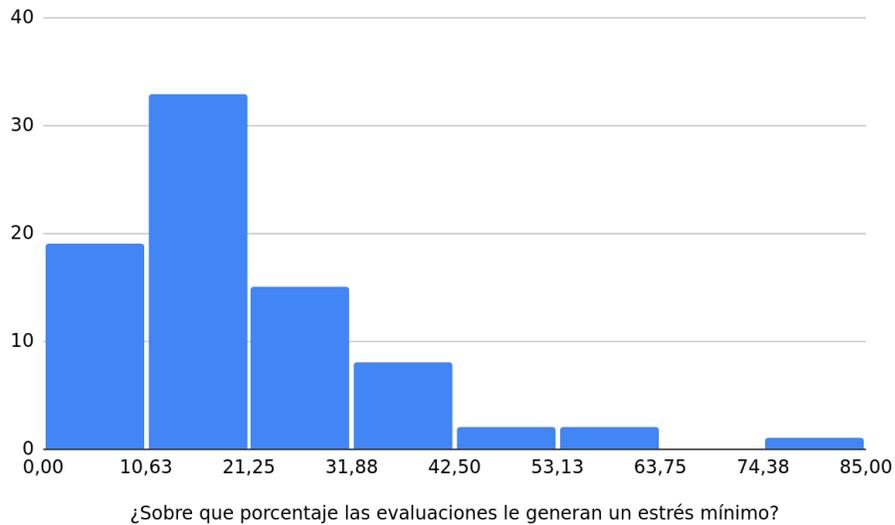


Figura 29. Ponderación total que causa bajo estrés durante la semana.

En la figura 30 se observan los resultados de la encuesta relacionados con el porcentaje de la ponderación total de la nota de evaluación, que le causa un estrés medio a los estudiantes, siendo el promedio de un 39.2 % y la mediana de un 40% del peso de la nota final de los cursos, para evaluaciones de una misma semana.

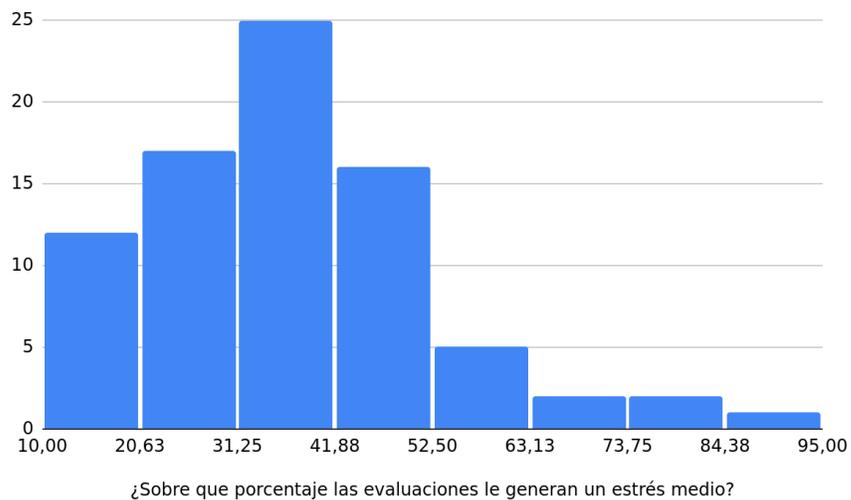


Figura 30. Ponderación total que causa estrés medio durante la semana.

En el gráfico de la figura 31 se observan los resultados de la encuesta referente al porcentaje de la nota final acumulada (ponderación total de evaluaciones) le causa un alto estrés a un estudiante, siendo el promedio de un 55.19% y la mediana de un 60% para evaluaciones presentes dentro de una misma semana.

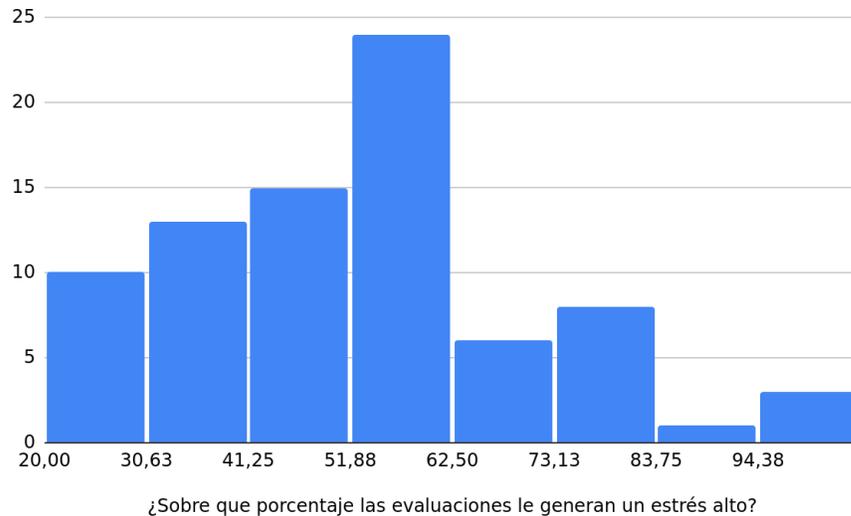


Figura 31. Ponderación total que causa estrés alto durante la semana.

#### 4.4.4.4 Análisis de resultados de la encuesta

Para representar la opinión estudiantil a través de la visualización de la carga académica, se decide modificar la definición de la misma, encontrándose al inicio solo asociada a la cantidad de evaluaciones, y ahora se relaciona además con la ponderación de la nota final de la evaluación.

Por lo siguiente, los rangos de dificultad diaria y semanal se modificaron en relación a la mediana obtenida en la encuesta anterior, de cuyos resultados se puede evidenciar que no se cuenta con un sesgo importante dentro de la misma. De esa manera, los niveles de carga académica quedan establecidos según el porcentaje de la nota final acumulada de las evaluaciones, y a su vez, estos niveles se verán incrementados si un estudiante posee 3 o más evaluaciones en un día, según los resultados de la primera parte de la encuesta. Los niveles en base a su ponderación acumulada son los siguientes:

1. Diario:
  - a. Bajo: 20%
  - b. Medio: 35%
  - c. Alto: 50%
  
2. Semanal:
  - a. Bajo: 20%
  - b. Medio: 40%
  - c. Alto: 60%

## 4.5 Servicio de autenticación de usuarios y asignación de roles

Para realizar la conexión con el servicio de autenticación de usuarios del del Portal DCC, fue necesario editar los modelos de datos para utilizar solo aquel “modelo de usuarios” asociado por defecto al framework Django, dado una limitación presente en el

sistema. Este requisito desencadenó en una reestructuración de los modelos de datos, modificando la forma en que se almacena la información de los usuarios.

En la figura 32 se puede observar el flujo de esta funcionalidad en el sistema, el cual se puede lograr de forma directa o a través del portal DCC. Una vez realizada la autenticación, se envía un token al frontend de la aplicación; si ésta proviene del portal DCC, entonces no se enviará mediante una respuesta del login, sino más bien mediante una redirección al front de UCalendar debido al comportamiento del portal, el cual redirige al servidor que requiere la autenticación.

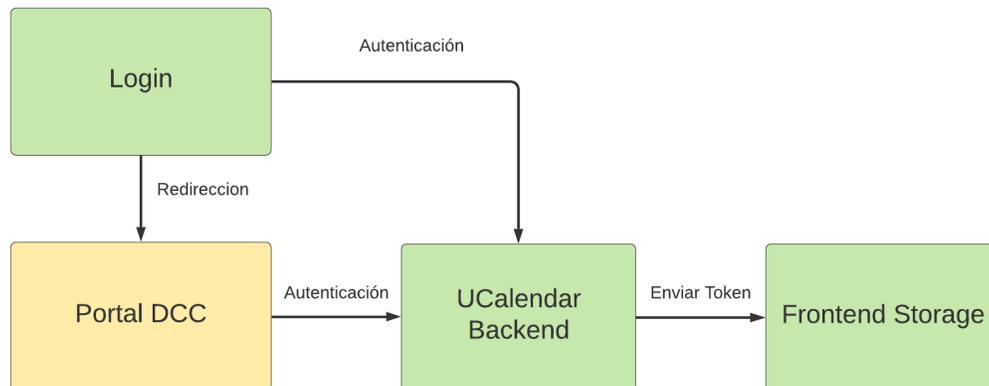


Figura 32. Flujo de autenticación de usuarios de UCalendar.

Dentro de la problemática principal, en lo que respecta a los roles, es necesario reestructurar aquellos permisos asociados en el sistema, dado que cierta información se encuentra a libre disposición sin importar el rol que exista dentro de éste. La separación de roles está mayormente manipulada por el frontend de la aplicación. Dado esto, los endpoints de la aplicación se modifican. A continuación se expone la interacción esperada al interactuar con las nuevas funcionalidades de la API.

#### 1. Cursos (“/api/cursos”):

- a. Crear (**POST** “/”): No implementado, estatus 400 siempre.
- b. Actualizar (**PATCH** “/:id”): Solo para administradores y profesores (este último del curso en cuestión), en otro caso retorna error con estado 401.
- c. Eliminar (**DELETE** “/:id”): No implementado, estatus 400 siempre.
- d. Leer (**GET** “/” o “/:id”): Se permite la lectura de cursos a todos los roles, siempre y cuando sean integrantes del curso en cuestión (exceptuando a administradores).
- e. Evaluaciones Sugeridas (**GET** “/:id/evaluaciones\_sugeridas/”): Se permite la lectura de las evaluaciones sugeridas de un curso a los

roles de administrador, profesor, siempre y cuando sean integrantes del curso en cuestión(exceptuando a administradores).

- f. Sugerir Evaluaciones(**POST** `"/:id/sugerir_evaluaciones/"`): Se permite sugerir las evaluaciones a los roles de administrador y profesor, siempre y cuando sean integrantes del curso en cuestión(exceptuando a administradores).

## 2. Semestres (`/semestres`):

- a. Crear (**POST** `"/`): No implementado, estatus 400.
- b. Actualizar (**PATCH** `"/:id"`): Solo para administradores, en otro caso de ser usado por otro rol retorna error con estado 401.
- c. Eliminar(**DELETE** `"/:id"`): No implementado, estatus 400 siempre.
- d. Leer(**GET** `"/` o `"/:id"`): Se permite la lectura de semestres a todos los roles, siempre y cuando sean integrantes del rol en cuestión o usuarios administradores.
- e. Evaluaciones(**GET** `"/evaluaciones_semestrales"`): Se permite la lectura de evaluaciones de un semestre específico a los roles administrador, profesor, auxiliar y ayudante.
- f. Mis Evaluaciones(**GET** `"/evaluaciones"`): Se permite la lectura de evaluaciones de un semestre específico a todos los usuarios que sean estudiantes en sus respectivos cursos.

## 3. Ramos (`/ramos`):

- a. Crear (**POST** `"/`): No implementado, estatus 400.
- b. Actualizar (**PATCH** `"/:id"`): Solo para administradores, en otro caso de ser usado por otro rol retorna error con estado 401.
- c. Eliminar (**DELETE** `"/:id"`): No implementado, estatus 400 siempre.
- d. Leer (**GET** `"/` o `"/:id"`): Se permite la lectura de ramos a todos los roles docentes y administradores, siempre y cuando sean integrantes del rol en cuestión o usuarios administradores.

## 4. Evaluaciones (`"/evaluaciones"`):

- a. Crear (**POST** `"/`): Administradores, profesores y auxiliares tiene permitido crear evaluaciones, en caso de los últimos dos roles solo se permite la creación en aquellos cursos en los que desempeñan el rol.
- b. Actualizar (**PATCH** `"/:id"`): Administradores, profesores y auxiliares tiene permitido actualizar evaluaciones, en caso de los últimos dos

roles solo se permite la actualización en aquellos cursos en los que desempeñan el rol.

- c. Eliminar(**DELETE** “/:id”): Administradores, profesores y auxiliares tiene permitido eliminar evaluaciones, en caso de los últimos dos roles solo se permite la creación en aquellos cursos en los que desempeñan el rol.
  - d. Leer(**GET** “/” o “/:id”): Se permite la lectura de evaluaciones a todos los roles, siempre y cuando sean integrantes de los cursos o administradores.
  - e. Aceptar Sugerencia Evaluaciones(**POST**“/:id/aceptar/”): Se permite aceptar las sugerencias de las evaluaciones a los roles de administrador, profesor y auxiliar siempre y cuando sean integrantes del curso en cuestión(exceptuando a administradores).
5. Notificaciones(“**notificaciones/config**”):
- a. Crear(**POST**): No implementado, estatus 400.
  - b. Actualizar(**PATCH** “/:id”): Actualiza información sobre plataformas de comunicación, y períodos de notificación para Usuarios estudiantes.
  - c. Leer(**GET** “/” o “/:id”): Todos los usuarios; sin embargo, no existen datos para usuarios no estudiantes.
  - d. Eliminar(**DELETE** “/:id”): No implementado, estatus 400.
  - e. Reset Token(**GET** “/:id/retoken”): Elimina el chat id de telegram asociado a un usuario, y resetea el token de conexión asociado a éste.

# Capítulo 5

## Evaluación de la solución

Para evaluar la solución planteada se usa el método de Recorrido Cognitivo desde el inicio del proyecto, esto con el fin de chequear constantemente la interacción con el sistema. Además, se realizan diversas validaciones al producto final, para validar su usabilidad y los puntos de mejora.

### 5.1 Descripción del proceso de evaluación

#### 5.1.1 Recorrido Cognitivo

Este método permite chequear cómo interactúan los usuarios con el producto en su primer contacto, y es usado para probar distintas funcionalidades de la aplicación en base a objetivos, donde el usuario debe ponerse un objetivo, buscar en la Interfaz las opciones para satisfacerlo, elegir la acción para progresar hacia su resolución, y finalmente realizar esta acción. Posterior a esto, se deben evaluar 4 puntos dentro de los cuales están los siguientes:

1. Correctitud de la solución
2. Visibilidad
3. Rotulación
4. Retroalimentación.

De esta manera, se realizan evaluaciones al completar las funcionalidades de forma evolutiva; lo que permite un progreso tanto en la evaluación de la usabilidad, como en la detección de errores indeseados. Finalmente, al término de la iteración se realiza una evaluación a 6 usuarios, distribuidos equitativamente entre distintos roles (entre estos están administradores, docentes y estudiantes).

Para hacer estas pruebas se sigue el siguiente flujo:

1. Entrar a UCalendar y loguearse con los datos de la cuenta entregada.
    - a. Escribir Keys
    - b. Loguearse en el sistema
  2. Comprender, y utilizar la vista inicial del sistema(Mapa de calor)
    - a. Encontrar un día con alta carga académica dentro del semestre.
    - b. Obtener información de las evaluaciones de esa fecha.
    - c. Obtener información de *fechas especiales* del semestre.
- *Para usuarios administradores:*

- d. Realizar operaciones CRUD de *fechas especiales*.
  - *Para usuarios solo docentes:*
    - e. Ver calendarios de distintos semestres de malla curricular
  - *Para Docentes y Estudiantes simultáneos:*
    - f. Visualizar tanto el calendario personal como el de docencia.
3. Comprender, y utilizar la vista de semestres
- a. Visualizar los cursos pertenecientes al usuario
  - b. Ver las evaluaciones pertenecientes a un determinado curso.
- *Usuarios Administradores:*
  - c. Edición de semestres.
- *Para usuarios Administradores y Profesores:*
  - d. Edición de Cursos.
- *Para usuarios Profesores y Auxiliares:*
  - e. CRUD de evaluaciones de uno de sus cursos.
  - f. Sugerir evaluaciones.
  - g. Aceptar alguna sugerencia.
4. Administradores: Comprender, y utilizar la vista de Evaluaciones específicas:
- a. Filtrar Evaluaciones por Semestre y/o Cursos.
  - b. Realizar operaciones CRUD.
  - c. Filtrar Evaluaciones con alertas de carga académica o *fechas especiales*.
  - d. Acceder a la información de evaluaciones con alertas.
5. Estudiantes: Comprender y utilizar la configuración de notificaciones:
- a. Configurar los periodos y medios en los que desea recibir las notificaciones.
  - b. Conectarse al bot de telegram de la aplicación.
  - c. Reiniciar el token de UCalendar.

## 5.1.2 Resultados del Recorrido Cognitivo

Como se menciona anteriormente se evalúa que el usuario cumpla con 4 objetivos asociados al funcionamiento del sistema: 1) la correctitud de la solución, 2) su estética, 3) la rotulación y 3) la retroalimentación. A continuación se evidencian los resultados obtenidos de los usuarios a los que se le aplicó el test.

1. Los usuarios logran el efecto esperado del login, notan la disponibilidad de éste en el sistema. Además, asocian esta funcionalidad con su efecto esperado, así como también visualizan el progreso que causa éste.
2. Los usuarios cumplen con los 4 objetivos, tanto para los casos 2.b y 2.c. Sin embargo, en el caso 2.a se presenta cierta dificultad de comprensión en la rotulación, en donde no se entiende totalmente en cómo se diferencia la carga académica, debido a una falta de una leyenda en el mapa de calor.
3. Los objetivos se cumplen en casi todos los casos, a excepción cuando se solicita editar las fechas del semestre actual, en donde no se comprende el significado del banderín que representa esto.
4. Los usuarios logran comprender y utilizar la vista de administración de evaluaciones, comprendiendo los efectos y causas de sus acciones.
5. Los usuarios logran en general realizar los efectos deseados en el sistema, y visualizar su progreso en éste. Sin embargo, se presentan ciertas dificultades en la rotulación del formulario, notando acciones no disponibles debido a la carencia de validación de semanas de notificación en el formulario.

## 5.2 Evaluación de la usabilidad de la aplicación

### 5.2.1 Descripción de la System Usability Scale

El enfoque de la System Usability Scale (SUS) [3] es evaluar la percepción del usuario sobre el sistema con respecto a 3 puntos:

1. *Usabilidad*: calidad de la experiencia de usuario en el sistema.
2. *Satisfacción*: percepción final del usuario posterior al uso de la aplicación.
3. *Simpleza*: facilidad en el uso del software.

De aquí se realizan 10 preguntas a los usuarios, siguiendo el modelo de respuestas de la escala Likert; con puntajes que van del 1 al 5, basado en el nivel de acuerdo con la afirmación. Las respuestas se indican con 1 cuando el evaluador está “muy en desacuerdo” y con 5 cuando está “muy de acuerdo”.

Para calcular el puntaje final de esta encuesta se utilizará la ecuación 3, la cual transforma los resultados a una escala [1,100], donde si el puntaje es mayor a 68 el resultado de la evaluación es considerado como positivo, según lo que establece la escala SUS.

$$PB = (P1 - 1) + (P3 - 1) + (P5 - 1) + (P7 - 1) + (P9 - 1) \quad (1)$$

$$PM = (5 - P2) + (5 - P4) + (5 - P6) + (5 - P8) + (5 - P10) \quad (2)$$

$$R = 2,5 * (PB + PM) \quad (3)$$

A continuación se enuncian los ítems (aseveraciones) perteneciente al cuestionario SUS, realizado a los 6 usuarios evaluadores:

1. Creo que me gustaría usar UCalendar frecuentemente.
2. Pienso que el sistema es innecesariamente complejo.
3. El sistema es fácil de usar.
4. Creo que necesitare asistencia de un técnico para usar el sistema.
5. Pienso que las funcionalidades del software están bien integradas.
6. UCalendar tiene demasiadas inconsistencias.
7. Creo que la mayoría de las personas aprenderán a usar el sistema rápidamente.
8. El sistema no es agradable para su uso.
9. Me siento seguro al usar el sistema.
10. Necesito aprender muchísimas cosas antes de lograr avanzar en el uso del software.

### 5.2.2 Resultados de la evaluación de usabilidad

En esta sección se muestran los resultados del cuestionario SUS (System Usability Scale) [3], se aprecian los puntajes que le asigna cada uno, los resultados finales y un promedio de cada uno de estos.

Según los resultados de la tabla se puede observar que el punto 5, referido a la correcta integración de las funcionalidades, es el que posee los puntajes más débiles de la encuesta. No obstante, dado que este resultado es aún considerado positivo, la mejora de este ítem no representa una problemática urgente de resolver en el sistema.

Dado que el puntaje final obtenido es de 90, se supera el requisito mínimo de 68 puntos para que los resultados fuesen considerados exitosos. Si bien la traducción de las preguntas pudieron haber ocasionado algún tipo de sesgo en los resultados, la diferencia entre los puntajes es lo suficientemente alta como para que esto no represente una amenaza real a los resultados.

Tabla 1. Resultados de la aplicación de la encuesta SUS

<b>Item / Usuario</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>Resultado</b>
Usuario 1	5	2	5	1	4	1	5	1	5	1	95
Usuario 2	5	1	5	2	4	1	4	1	5	1	92.5
Usuario 3	3	1	5	2	4	1	4	2	4	2	80
Usuario 4	4	2	5	1	4	1	5	1	5	2	90
Usuario 5	5	2	5	1	5	2	4	1	5	2	90
Usuario 6	4	1	4	1	5	1	5	2	5	1	92.5
<b>Promedio</b>	<b>4.5</b>	<b>1.33</b>	<b>4.83</b>	<b>1.33</b>	<b>4.33</b>	<b>1.16</b>	<b>4.5</b>	<b>1.33</b>	<b>4.83</b>	<b>1.33</b>	<b>90</b>

## Capítulo 6

### Conclusiones y trabajo a futuro

Tal y como hemos podido comprobar en los capítulos anteriores, este trabajo de título se enfocó en modificar la última iteración de UCalendar, tanto para dotarlo de las herramientas mínimas necesaria para su paso a producción, como para lograr un uso amigable del sistema para sus distintos roles y su experiencia durante el uso del mismo fuese más provechosa.

Es en la etapa de evaluación del sistema donde se pudo comprobar que, si bien aún existen funcionalidades a mejorar (como la integración de funcionalidades y ciertos problemas de rotulación), éstas no impiden el correcto uso del sistema, lo que hace notar que las funcionalidades implementadas resultan amigables para los usuarios.

El motivo de la realización de este trabajo de memoria fue el poder ofrecer soluciones que permitan facilitar y mejorar el proceso de gestión de carga académica al DCC, y así beneficiar directamente a los alumnos, académicos y funcionarios de la unidad. Es por esto que funcionalidades como la extracción de datos externos y los cambios de visualización en el mapa de calor, resultaron ser las funcionalidades más relevantes del sistema, ya que generan grandes cambios en cómo los usuarios interactúan con el mismo.

Debido a los múltiples cambios que podrían ser realizados en el sistema en el futuro, se debió actualizar el uso de múltiples tecnologías. Esto derivó en decisiones importantes de refactoring y rewriting de código, que permitieron cumplir con objetivos de robustecimiento y escalabilidad de la aplicación, así como en ahorros importantes en costos de desarrollo. Sin embargo, debido a que algunas decisiones se tomaron de manera algo tardía, este ahorro no logró ser maximizado.

Con respecto al trabajo futuro, cabe decir que existen varias tareas a realizar, no obstante, es de suma importancia priorizarlas correctamente. Por lo anterior, una vez que el proyecto se encuentre en producción, se deben identificar las necesidades de los usuarios que no se encuentran satisfechas, en conjunto con las principales falencias de éste. Así también se debe atender a nuevas propuestas, como las que surgieron de las evaluaciones del sistema. Por ejemplo, la integración de la aplicación con Google Calendar, para que los alumnos pudiesen exportar el calendario. Además, se sugirió extender el mapa de calor, con funcionalidad que permita evitar (o generar alertas de) topes de horario de los integrantes de comisiones de memorias o tesis.

# Bibliografía

- [1] Documento Histórico del Proyecto UCalendar. CC5401: Ingeniería de Software II. DCC, FCFM, Universidad de Chile. Semestres Otoño y Primavera 2020.
- [2] Aslam, Mubeen M.: Are you selling the right colour? A cross-cultural review of colour as a marketing cue 2005, 1-14.
- [3] Brooke, J. SUS: A “quick and dirty” usability scale. In Jordan, P.W., Thomas, B., McClelland, I.L., & Weerdmeester, B. (Eds.). Usability Evaluation In Industry (1st ed.). (pp.189-194). CRC Press. (1996). <https://doi.org/10.1201/9781498710411>.

# Glosario

Calendario: Visualización de las evaluaciones de un conjunto de cursos de un semestre, distribuidas por día. En específico, la visualización corresponde a un mapa de calor.

Curso: Instancia específica de un ramo, es decir, un ramo asociado al semestre en el cual se dicta y su respectiva sección.

DCC: Departamento de Ciencias de la Computación de la FCFM de la Universidad de Chile.

Evaluación: Pertenece a un curso, y es toda actividad que lleve una nota y deba ser realizada por los alumnos del respectivo curso.

FCFM: Facultad de Ciencias Físicas y Matemáticas (FCFM) de la Universidad de Chile.

Fechas Especiales: Fechas no lectivas para la FCFM. Tales como vacaciones, feriados u otros. Se incluyen las semanas en las que no hay clases como la semana del 18 de septiembre o la semana olímpica (que no debe haber evaluaciones).

Ramo: Corresponde a la existencia de un curso, pero no la instancia específica del mismo. Sólo se refiere al nombre del ramo y a que semestre corresponde en específico (en la malla curricular).

Rol: Función que desempeña un usuario en la aplicación y en cada curso.

Semestre: Periodo en que se dictan los cursos.

Usuario: Persona que utiliza la aplicación, y tiene un perfil asociado.

Secretaría Docente: Equipo de funcionarios/as que se encarga de gestionar los distintos aspectos relacionados a la docencia en el DCC.

Carga Académica: Conjunto de instancias evaluativas o asignaturas que una persona adquiere en un contexto estudiantil.